



WebRTC: Video
Over The Internet



The Five Pillars
Of Hadoop

₹ 100

OpenSource

Volume: 04 | Issue: 01 | Pages: 108 | October 2015

THE COMPLETE MAGAZINE ON OPEN SOURCE **ForYou**

An **EFY** GROUP Publication

A PEEK AT THE LATEST PROGRAMMING LANGUAGES

- Getting Started With Dart
- Go: The Simple Programming Language
- JavaScript - The New Parts

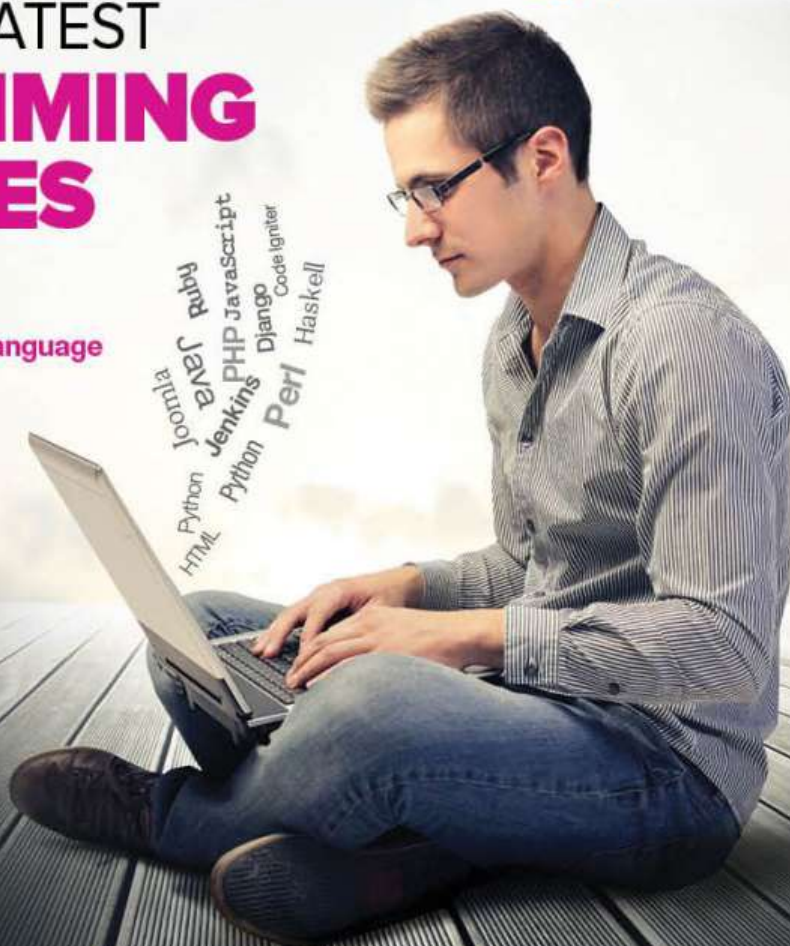
Python
HTML
jQuery
Java
Ruby
PHP
JavaScript
Jenkins
Django
Code Igniter
Perl
Haskell

Case Study:

Open Source Deployments At
redBus.in

An Interview With

Sanjay Deshmukh,
GM, Business Mobility, VMware, APJ



The Community Is Meeting The Industry!

Have You Marked The Dates On Your Calendar?

12th Edition

OPEN
SOURCE INDIA

NIMHANS Convention Center

19 - 20
November
2015

BENGALURU

Asia's largest conference on Open Source

enabling futurability



An all-in-one comprehensive
Data Center Management Suite.

Simplicity
at its Best.



Try it, To Believe it !

✕ Inventory and Asset
Management

⚙️ Fully Integrated Workflow
Management

☁️ Hybrid Cloud Management -
Orchestration

♻️ ITIL Framework Compliant
Management System

🖥️ Intense Monitoring for power
Network & others

🔔 Instant Notifications and Alerts
via. Emails, Sms & Irc channels

Call Now:

1800 209 3006

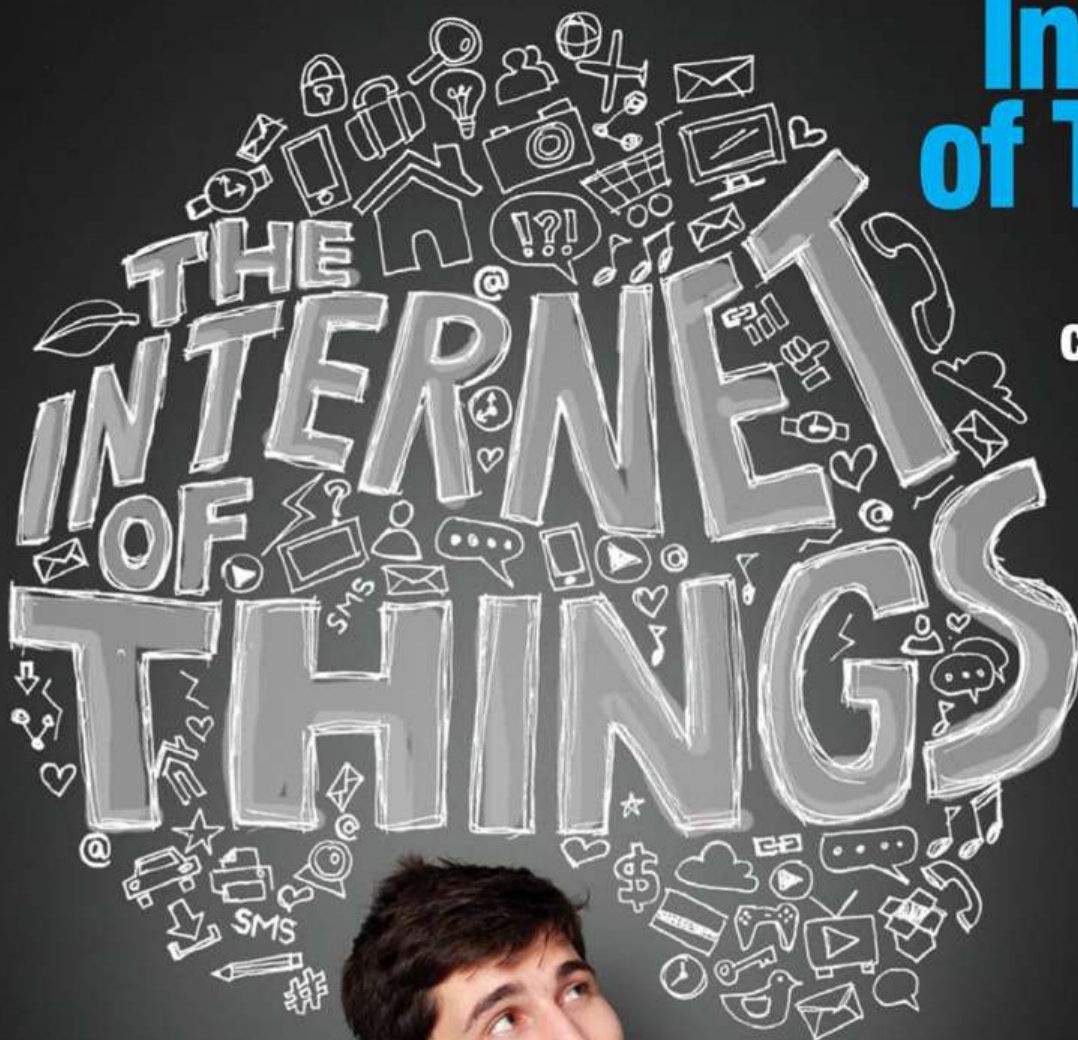
esds 
enabling futurability

Mumbai | Delhi | Bengaluru | Nashik | Leeds (UK)

ESDS SOFTWARE SOLUTION PVT. LTD.

Toll Free No. : 1800 209 3006 | Fax : +91 95 95 247 247

Email : relationship@esds.co.in | Website : www.esds.co.in



Bangalore International Exhibition Centre (BIEC), Bengaluru

January 11th-13th, 2016
Bengaluru

Contents

ADMIN

- 25** Malware Analysis Using REMnux
- 33** Openjudge: An Online Judge for Offline Programming Contests
- 37** Chef: Recipes that Turn Infrastructure into Code
- 41** The Five Pillars of Hadoop
- 43** WebRTC: The Way Forward for Video Over the Internet

DEVELOPERS

- 54** Android Studio: A Platform for Android Development
- 57** Using Python for the Cloud and Big Data Analytics
- 66** Building a Music Player Application in App Inventor 2
- 68** Getting Started with Dart
- 71** JavaScript: The New Parts
- 74** An Easy-to-Follow Git Based Development Workflow
- 78** The Rise of Mobile Application Development Platforms
- 82** Why Are Organisations Moving to Microservices?

OPEN GURUS

- 86** OpenModelica: A Powerful Engineering Modelling and Simulation Tool



48 Building Apps with Meteor



62 Go: The Simple Programming Language

REGULAR FEATURES

- | | |
|--------------------------|------------------------------|
| 08 New Products | 47 Editorial Calendar |
| 10 You Said It... | 104 Tips & Tricks |
| 11 FOSSBytes | |

#CGCINDIA



29th - 31st Oct 2015
BIEC, Bengaluru

Everybody's talking about
digital transformation.

Now hear it from those who matter.



Theme:

D!conomy - 'The Digital Economy'

How digital technologies will impact
our businesses, lifestyle and workplace.

3 days, 30 power packed sessions,
over 70 speakers including leaders,
innovators, business and government
heads will set the tone for digital
transformation in India.

Focus on Digital Transformational Leaders,
Future of Work, 4G Enabled Enterprise, Digital
Banking, and Cyber Security.

Early Bird Offer 10% Off till 15th September

Register now:

Mr. Mohd. Farooq - CGC Delegate Program Lead

+91-9004691833

www.cebit-india.com



Deutsche Messe



FIERA MILANO

CeBIT
INDIA

Contents



- 23** “From a personal preference perspective, I would choose an open source solution, any day”
—Anoop Menon, CTO, redBus.in



- 101** “Business mobility is changing the core Business processes”
—Sanjay Deshmukh, general manager, business mobility, VMware, APJ

- 88** Python and pcap in ns-3

- 93** Using GNU Emacs as an Internet Relay Chat

FOR U & ME

- 17** Open Source Technology: First Love of the Leaders at AskMeBazaar.com!

- 96** Are Business Intelligence Tools Really Intelligent?

- 98** An Introduction to Open Source Programming Languages

COLUMNS

- 21** CodeSport

Buyer's Guide

- Things to Look at While Choosing an IaaS Provider **19**

DVD Of The Month

- Here are some useful Linux distributions for advanced users. **106**

Editor

RAHUL CHOPRA

Editorial, Subscriptions & Advertising

DELHI (HQ)

D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020

Ph: (011) 26810602, 26810603; Fax: 26817563

E-mail: info@efy.in

Missing Issues

E-mail: support@efy.in

BENGALURU

Ph: (080) 25260394, 25260023

E-mail: efyblr@efy.in

Customer Care

E-mail: support@efy.in

Back Issues

Kits 'n' Spares

New Delhi 110020

Ph: (011) 26371661, 26371662

E-mail: info@kitsnspares.com

Advertising

CHENNAI

Ph: (044) 42994363

E-mail: efyeng@efy.in

HYDERABAD

Ph: (040) 67172633

E-mail: efyeng@efy.in

KOLKATA

Ph: (033) 22294788

E-mail: efyeng@efy.in

MUMBAI

Ph: (022) 24950047, 24928520

E-mail: efymum@efy.in

PUNE

Ph: 08800295610/09870682995

E-mail: efypune@efy.in

GUJARAT

Ph: (079) 61344948

E-mail: efyahd@efy.in

JAPAN

Tandem Inc., Ph: 81-3-3541-4166

E-mail: tandem@efy.in

SINGAPORE

Publicitas Singapore Pte Ltd

Ph: +65-6836 2272

E-mail: publicitas@efy.in

UNITED STATES

E & Tech Media

Ph: +1 860 536 6677

E-mail: veroniqueamarque@gmail.com

CHINA

Power Pioneer Group Inc.

Ph: (86 755) 83729797, (86) 13923802595

E-mail: powerpioneer@efy.in

TAIWAN

J.K. Media, Ph: 886-2-87726780 ext. 10

E-mail: jkmedia@efy.in

Exclusive News-stand Distributor (India)

IBH BOOKS AND MAGAZINES DISTRIBUTORS LTD

Unit No.10, Bezzola Complex, Opp. Suman Nagar,

Sion Trombay Road, Chembur,

Mumbai - 400 071

Phones : 022 - 40497401/02

E-mail: info@ibhworld.com

Printed, published and owned by Ramesh Chopra. Printed at Tara Art Printers Pvt Ltd, A-46/47, Sec-5, Noida, on 28th of the previous month, and published from D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020. Copyright © 2015. All articles in this issue, except for interviews, verbatim quotes, or unless otherwise explicitly mentioned, will be released under Creative Commons Attribution-NonCommercial 3.0 Unported License a month after the date of publication. Refer to <http://creativecommons.org/licenses/by-nc/3.0/> for a copy of the licence. Although every effort is made to ensure accuracy, no responsibility whatsoever is taken for any loss due to publishing errors. Articles that cannot be used are returned to the authors if accompanied by a self-addressed and sufficiently stamped envelope. But no responsibility is taken for any loss or delay in returning the material. Disputes, if any, will be settled in a New Delhi court only.

SUBSCRIPTION RATES

Year	Newstand Price (₹)	You Pay (₹)	Overseas
Five	6000	3600	—
Three	3600	2520	—
One	1200	960	US\$ 120

Kindly add ₹ 50/- for outside Delhi cheques.

Please send payments only in favour of **EFY Enterprises Pvt Ltd**. Non-receipt of copies may be reported to support@efy.in—do mention your subscription number.

ALIVE

'I SWEAR, I'VE SEEN THAT SOFA IN MY DREAM!'

RELAX! JUST SCAN IT AND GET IT.



INTRODUCING

OneScan

SEARCH WAS NEVER SO EASY!



'NOW WHEN YOU SEE A PRODUCT YOU WANT TO BUY JUST SCAN IT WITH 'ONESCAN''

- FIND BEST DEALS
- COMPARE PRICES



SCAN WHAT YOU LIKE



KNOW ITS DETAILS



FIND OUT WHERE TO GET IT

Download app from Aliveonescan.com/app | Available only on Android

NEW PRODUCTS

Portable photo printer from Micromax

Indian handset manufacturer Micromax has unveiled its new portable photo printer, dubbed the YUPIX, under its YU series comprising affordable high-end smartphones like YU Yureka and YU Yuphoria.

The device is capable of converting digital memories from your smartphones to hard copies measuring 5.33cm x 8.63cm (2.1"x 3.4") in just 60 seconds.

The image transfer speed of the printer is 10 times faster than Bluetooth.

The printer has a resolution of 291 dpi, supports 16.7 million colours and is compatible with both Android and iOS smartphones. Powered by a 750mAh battery, the printer's dimensions are 75.9cm x 152.6cm x 23.8cm (2.99"x 6.01"x 0.94") and it has a net weight of 273gm, including the battery and cartridge.

With an integration of dye sublimation technology, the printer is said to include a cartridge that does not require a cleaning roll. The YUPIX printer also comes with a companion application, which can be downloaded from Google Play Store to wirelessly connect smartphones via Wi-Fi or NFC.

The Micromax YUPIX is available in blue via Amazon.in.

Address: Micromax Informatics Limited, 90B, Sector-18, Gurgaon 122015; **Ph:** +91-124-4811000; **Email:** info@micromaxinfo.com



Price:
₹ 6,999

Micromax introduces mid-range 4G smartphone

Micromax has recently added a new smartphone to its mid-range segment, the Canvas Nitro 4G. The dual SIM smartphone sports a 12.7cm (5 inch) HD (720x1280) IPS screen with oleophobic protection on it and Corning Gorilla Glass 3.

The smartphone runs on Android 5.0.2 Lollipop and is powered by the 1.4GHz Snapdragon 415 octa-core processor. It is equipped with 2GB of LPDDR3 RAM and 16GB of inbuilt storage, which can be expanded up to 32GB via microSD card.

The Canvas Nitro 4G comes with a 13 megapixel rear auto-focus camera, with dual-LED flash and a 5 megapixel fixed-focus front-facing camera along with a 2500mAh battery.

Apart from 4G LTE connectivity, the smartphone offers Bluetooth 4.0, Wi-Fi, microUSB and a 3.5mm audio jack.

The Canvas Nitro 4G is available at all retail and online stores.

Address: Micromax Informatics Limited, 90B, Sector-18, Gurgaon 122015;

Ph: +91-124-4811000; **Email:** info@micromaxinfo.com



Price: ₹ 10,999

Portable power bank with dual USB

One of the leading global suppliers of laptop cases and accessories, Targus, recently released a compact, lightweight and portable power bank called the Targus APB031AP.

The power bank is a must-have travel accessory on account of its ultra-thin and compact size, combined with its fast-charging capabilities. It comes with dual 2.1A output USB ports with 8400mAh power. Equipped with Samsung cells that offer reliable battery performance, it has the ability to charge a smartphone five times and an iPad with additional 10 hours, on a single charge. It allows users to charge their smartphones while using them. The power bank has a three-level LED battery indicator to show the remaining power status.

The Targus APB031AP is available via Amazon.in.

Address: Targus Group International Inc, #30, SF-1, 2nd Floor, Sapthagiri, 10th Cross, 15th Main Road, RMV Extension, Sadashivnagar, Bengaluru 560080; **Website:** www.targus.com/in/



Price: ₹ 1,499

Motorola's new phone is a 'perfect partner'

Multinational telecommunications company, Motorola, has launched Moto X Play, which comes with supposedly 30 hours of battery life, making it what the company claims is 'the perfect partner'.

The 13.9cm (5.5") display smartphone comes with a display of full HD (1080 p) screen resolution. Powered by a 64-bit Qualcomm Snapdragon 615, the smartphone features a 1.7GHz processor with 2GB RAM. It runs on Android 5.1.1 Lollipop and also features 4G LTE support.

The device sports a 21MP rear and 5MP front-facing camera for perfect selfies. A much highlighted feature of the phone is its staggering 3,630mAh battery, which enables it to be used for more than 30 hours.

The dual SIM smartphone has been released with 16GB and 32GB variants in India, both of which are expandable via microSD card.

The connectivity options of Moto X Play include Bluetooth 4.0LE and Wi-Fi-802.11 a/g/b/n (dual band capable). The Moto X Play base models are available in black and white, along with customised colours, via online and retail stores.



Price: ₹ 18,499 and ₹ 19,999 for 16GB and 32GB, respectively

Address: Motorola Solutions, Sector 14, Mehrauli Gurgaon Road, Gurgaon 122001; **Ph:**(91)-124-2303212, 4192000; **Website:** www.motorolasolutions.com

Bose unveils wireless headphones

Best known for its audio products, Bose has recently launched its new wireless headphones in India, called Bose Soundlink Headphones 2.

They come with the latest Bluetooth technology, which lets you easily connect to your mobile devices with seamless audio/video sync, along with TriPort technology and Active EQ, which delivers sound that's crisp and powerful at any volume.

The headphones feature technology that automatically adjusts the call volume according to the environment. The company claims they are extremely comfortable and can run for up to 15 hours once fully charged. Featuring impact-resistant materials, glass filled nylon and corrosion-resistant stainless steel, they are equipped with NFC connectivity which enables users to easily switch between calls and music. Other features include active equalisation, an advanced microphone system and high definition voice.

The headphones come with voice prompts which inform the user as to who is calling, what device they are connected to and also the remaining battery life.

The Bose Soundlink Headphones 2 can be folded flat; hence, they are easy to carry. Available in black and white, they can be purchased online and at retail stores.



Price: ₹ 21,150

Address: Bose Corporation India Private Limited, 3rd Floor, Salcon Aaurum, Plot No.4, Jasola District Centre, Mathura Road, New Delhi 110025; **Ph:** +911143080200, 43080232; **Website:** <http://www.boseindia.com/>

Lapcare's attachable Bluetooth speakers

Lapcare, a leading brand in IT products and smartphone/tablet accessories, has added Bluetooth speakers under the Lapcare YO! Lifestyle series, called Klik speakers.

Apart from being portable, the speakers come with a built-in carabiner, which allows music lovers to attach the speakers nearly everywhere, including a backpack, a purse, etc.

The tiny but stylish Klik speakers offer amazingly loud and clear music, on the go. They are made of a durable material, along with a colourful rubber finish and a unique barrel shaped design.



Price: ₹ 1,499

The speakers are compatible with nearly all Bluetooth devices including smartphones, tablets, etc, with a range of about 10 metres.

They come with a 3.5mm audio cable and a MicroUSB charging cable. They are available in four lively colours: cyan, pink, lemon yellow and light green, via all e-commerce sites including Flipkart, Snapdeal, Amazon and eBay.

Address: Lapcare India Pvt Ltd, 601 Bhandari House, 91 Nehru Place, New Delhi 110019; **Email:** customercare@lapcare.com; **Website:** www.lapcare.com



YOU SAID IT



Why has the 'Open Gadgets' section been discontinued?

I noticed that in recent months the 'Open Gadgets' section has been removed from OSFY. It was helpful to me while selecting smartphones. Can you share the reason why this section has been discontinued?

—Chandramohan,
rkchandramohan@gmail.com

ED: Thanks a lot for the feedback. It really makes us feel good when readers benefit from the contents of the magazine. We noticed that there are several online websites with tools to compare smartphone features. Comparing phones using such tools is much easier and effective. Having details of smartphones in a print magazine can be informative but does not suffice as a buying tool. So, we decided to discontinue the section and use those pages for technical content. Do keep writing to us; our readers' suggestions help us to improve the quality of the magazine.



Which distribution is the best?

I remember your magazine from the very beginning, when it was called LFY. In fact, I have a huge collection of early issues of LFY. Now, it is called OSFY and will soon complete three years under this new masthead. Strangely, though I like and read OSFY, I have not been able to install any distro. However, I did get a taste of Linux via the live CDs/DVDs that come bundled with the magazine. With the vast choice of distros available, I can't figure out which would be the best to install. Could you suggest one that is easy to install and is future-safe. Have I to uninstall Ubuntu 15.4 if I want to update it to Ubuntu 15.10?

Could I request you to cover Ubuntu 15.4 installation, like you did in the case of Fedora 22 in the July issue of OSFY?

—Goswami Prashnata,
pacificg@gmail.com

ED: It really makes us feel good to know that you have been such a loyal reader. It is true that there are too many Linux distros available, which makes it really difficult for users to decide which one to use. Each distro has its own benefits. It totally depends on a user's specific requirements. The reason behind having a live DVD is this—it allows you to test and evaluate the distro and see whether it meets your needs or not. You can install the right Linux distro after that.

Yes, you can upgrade Ubuntu 15.4 to 15.10. You can refer to the official documentation for this. We will surely try to implement the suggestions given by you. We love to hear from our readers about the contents of our magazine and try our best to match their expectations. So keep sending us your views on OSFY! **END** 



Facebook Feedback



Sanoop Sam Sathish: I am unable to subscribe to your magazine from magzter using my debit card. Do let me know if any other options are available.



Open Source For You: If you want to read the magazine online, you can subscribe to the ezine version of OSFY at <http://ezines.efyindia.com/>



Thanks.



Sanoop Sam Sathish: When can we expect the Kali Linux distro with the magazine?



Open Source For You: Watch out for the October 2015 issue. We will be bundling the Kali Linux 2.0 live edition.



Please send your comments or suggestions to:

The Editor,
Open Source For You,
D-87/1, Okhla Industrial Area, Phase I,
New Delhi 110020, **Phone:** 011-26810601/02/03,
Fax: 011-26817563, **Email:** osfyedit@efy.in

The final version of Linux Kernel 4.2 now ready for download

The final version of Linux kernel 4.2 is now available for download. Linus Torvalds recently announced the release of the widely used operating system kernel. Linux kernel 4.2, which is considered to be one of the biggest kernels currently available, can now be downloaded directly from

kernel.org. “So judging by how little happened this week, it wouldn’t have been a mistake to release 4.2 last week after all, but hey, there’s certainly a few fixes here, and it’s not like delaying 4.2 for a week should have caused any problems either. So here it is, and the merge window for 4.3 is now open. I already have a few pending early pull requests, but as usual, I’ll start processing them tomorrow and give the release some time to actually sit,” announced Torvalds.

The latest version of the Linux kernel comes with many interesting features and fixes. It brings Intel Assembly x86 code rewrites, encryption of F2FS per-file, handling of NCQ TRIM, ARM board, Jitter RNG improvements, queue spinlocks, an all-new AMD GPU driver, along with various other updated drivers and performance improvements.

Meanwhile, developers can compile the latest version of Linux kernel on their GNU/Linux distribution. People who aren’t sure of how it works are recommended to wait for their vendors to upgrade the operating system kernel to version 4.2, which is expected to happen within a few months.

Maintainers can download the newest version and try to compile, test and eventually upgrade the current kernel packages of their distributions. The final version of Linux kernel 4.2 promises to provide a consistent system supporting new hardware components.



The Indian government launches Debian based BOSS to replace Windows

After Microsoft Windows hit the headlines for its privacy issues, the Indian government reportedly decided to switch to a new operating system with



powerful security features. Dubbed as Bharat Operating System Solutions (BOSS), the government’s own operating system has been developed by the Centre for the Development of Advanced Computing (C-DAC). Created for official purposes, BOSS is a Linux based distribution built with the help of DRDO. “We have no dearth of developers here. BOSS has almost all the features that one can get in, say, Windows. The earlier version was less user-friendly and had few features. We will seek the help of Indian software biggies to develop it further,” confirmed an official. Reports

Italy’s Ministry of Defence to drop Microsoft Office and switch to LibreOffice

Recently, it was reported that Munich has ditched Windows and adopted Linux. Now we hear that Italy’s Ministry of Defence is all set to abandon Microsoft Office and switch to LibreOffice instead. Reports further suggest that LibreItalia Association has joined hands with the ministry to give wings to the movement. Apparently, a treaty has been signed between Sonia Montegiove, president, LibreItalia Association and Ruggiero Di Biase, general executive manager, Automated Information Systems, the Ministry of Defence.



“Under the agreement, the Italian Ministry of Defence will develop educational content for a series of online training courses on LibreOffice, which will be released to the community under Creative Commons, while the partners, LibreItalia, will manage voluntarily the communication and training of trainers in the ministry,” said Italo Vignoli, honorary president of LibreItalia.

As LibreOffice is open source software, adopting this application would free the ministry from proprietary software. Government departments of many other European countries like France, Germany, Spain and the United Kingdom have already switched to LibreOffice and found it better than Microsoft Office. Developed by The Document Foundation, LibreOffice is a free and open source office suite.

BaruwaOS 6.7 drops CentOS and adopts Red Hat Enterprise Linux 6.7

The final version of BaruwaOS has finally been released with various features and fixes.



With installable ISO images, which support 32-bit as well as 64-bit hardware

architectures, the latest version of Baruwa Enterprise Edition 6.7 is available for download through its official website. Meanwhile, users working on BaruwaOS 2.0.7 or the later version of the operating system can use the *baruwa-setup* command-line utility to update their version. Baruwa Enterprise Edition 6.7, which was previously based on the CentOS distro, now uses the Red Hat Enterprise Linux (RHEL) 6.7 platform. While CentOS was also based on RHEL's source code, the two distros apply different minor patches at times.

The latest version of Baruwa comes as an updated Baruwa-Setup package along with the substantial update for the Baruwa-Puppet package. "Today, we are issuing updated BaruwaOS 6.7. This update tracks the upstream OS release 6.7. Packages updated in the upstream have been rolled into BaruwaOS 6.7. The release also includes Baruwa package release 2.0.9," reads the official announcement.

In addition to the change in the platform, many packages on the Baruwa operating system have been updated. These include MailScanner 4.85.5-9, SpamAssassin 3.4.1-3, Exim 4.84.12, Uwsgi 1.4.10, GnuTLS 3.3.16-1 and Compat-GnuTLS 2.8.5-1. Various other packages have also been updated, which include some upstream software repositories of RHEL 6.7.

further say that the Debian based BOSS uses 3.14 GNOME as a desktop. The enhanced model of the OS relies on desktop environment version 3.4 and features the latest kernel 3.16. It is expected to support Intel's 64-bit/32-bit architecture, the 32-bit Intel GPU and a 3D desktop.

Initiated by National Resource Centre for Free and Open Source Software (NRCFOSS) of India, BOSS claims to support 18 Indian languages, including Marathi, Assamese, Urdu, Kannada, Konkani, Malayalam and Sanskrit, and has been planned to be rolled out by the end of September 2015.

Apart from BOSS, the government is also planning to launch EduBOSS, a user-friendly Linux OS with educational applications targeting school kids.

Interestingly, India is not the only country that is switching to Linux; many other countries like Italy and Germany also want to stop being dependent on Windows.

Rackspace platform now supported by Ubuntu Linux

With an aim to offer the best cloud experience to its users, Canonical's Ubuntu Linux is all set to support the Rackspace platform. Udi Nachmany, the man behind the Ubuntu Certified Public Cloud programme, announced the availability of Rackspace support. While ensuring that the latest editions are always at their disposal, maintainers might soon develop the Rackspace images on the Ubuntu Certified Public Cloud infrastructure on a regular basis. While Ubuntu Linux images are expected to be circulated via Ubuntu Certified Public Cloud (CPC), Rackspace customers are supposed to get its Fanatical support. "The reason our customers choose to run Ubuntu is to get things done, quickly, easily and without worry. The less time they spend thinking about and maintaining the platform they're running on, and the more time they can spend on their core business or mission, the happier we are," said Udi Nachmany.

As Canonical and Rackspace join hands, tech enthusiasts would soon get to see many interesting features and fixes, which include the former's support across all Rackspace platforms. If all goes as anticipated, users would soon be able to connect private as well as public clouds with dedicated hardware.

Reports further suggest that OpenStack's co-founder is in talks with Ubuntu Linux for OpenStack deployments. The two firms may soon enter into a tie-up.

The world's most popular free operating system is now spreading its wings to include cloud technology as well. Now that's what we call a true Ubuntu strategy!

New stable version of Manjaro Linux launched

Manjaro developers have announced the new stable version of the Manjaro



Linux operating system. With the addition of the MATE 1.10.2 desktop environment, Linux kernel 4.2, Nvidia 355.11, Calligra 2.9.7 and systemd 225, Manjaro Linux is now ranked as one of the most modern Linux distributions

currently available. While Linux 4.1.6-3 LTS, Linux 3.18.21 LTS, Linux 3.19.8.6 and Linux 3.13.11.26 kernel packages have been added, Linux kernel 4.0.9 has been eliminated for being out-of-date. Apart from these updated

kernel packages, it also comprises Arch Linux software repositories with new features and fixes.

“It is always good to check for updated packages, even if we don’t notify you about them. For notifications on a daily basis, please take a look at our mailing list. Please give us feedback and report any findings with this update,” said Philip Müller, Manjaro Linux developer.

All Manjaro Linux 0.8.13 and 0.8.13.1 users are recommended to update their systems so that they could enjoy the new features and fixes. The operating system supports 3.10.87 LTS, 3.12.47 LTS, 3.13.11.26, 3.14.51 LTS, 3.16.7.16, 3.18.21 LTS, 3.19.8.6, 4.1.6 LTS and 4.2.0, and users can choose any of the mentioned kernel packages.

‘Hack Codegen’, software that writes code, is now open source

In a recent initiative that supports open source technologies, Facebook has decided to open source Hack Codegen, which is software that’s designed to write code. In a blog post, Facebook explained the motivation behind creating Hack Codegen, saying, “Before Hack Codegen was developed, we mainly generated code through concatenating strings and a few helper functions. On the product infrastructure team at Facebook, we had been looking into how to improve one of our internal systems for reading and writing data.”

Making an announcement about open sourcing Hack Codegen, Facebook shared, “Hack Codegen is a library for generating Hack code and writing it into signed files that prevent undesired modifications. Being able to generate code through automated code generation allows programmers to increase the level of abstraction by making frameworks that are declarative and that are translated into high-quality Hack code. We’ve been using Hack Codegen at Facebook for a while. After seeing so much internal success, we open sourced this library so that more people could take advantage of it.”

Hack code is Facebook’s programming language developed for HipHop

Virtual Machine (HHVM).

According to the official website, Hack is a programming language for HHVM. Hack reconciles the fast development cycle of a dynamically typed language with the discipline provided by static typing, while adding



many features commonly found in other modern programming languages.

Hack provides instantaneous type checking by incrementally checking your files as you edit them. It typically runs in less than 200 milliseconds, making it easy to integrate into your development workflow without a noticeable delay.

Before Hack Codegen was created, Facebook used to generate code via concatenating strings. The social media company later discovered that this technique was not good enough to scale up. Facebook wrote, “We realised early on that we would need a good library to generate code. At the time, we didn’t do that much code generation at [Facebook], mostly dumping values into arrays, so we didn’t have any good tools except for signing files. This is the need that motivated us to write this library.”

Due to the organic adoption and homegrown refactoring tools, Facebook migrated almost all of its PHP codebase to Hack over a year-and-a-half ago.

Ubuntu Touch now comes with Wi-Fi transfer app

Ubuntu Touch has already impressed tech enthusiasts across the world. In fact, it is being touted as the next hot smartphone operating system. Recently, developers have added one more interesting application in its line of apps. Now, you can find the WifiTransfer app in the Ubuntu store with new features. While the app repository for Ubuntu Touch may not be impressive, it does contain a few rare apps which cannot be found anywhere else. WifiTransfer is one such application, which enables users to transfer files between a phone and



an OS in the same network using the FTP protocol. Developed by Stuart Langridge, a dedicated member of the community, the app has been successful in grabbing a lot of eyeballs.

“I have updated my WifiTransfer app for Ubuntu phones (it lets you copy files, documents, etc, onto the phone from your machine over Wi-Fi without needing a cable) to add two useful features. First, you can now browse the files you’ve copied to the phone from within the app, which means that it’s much easier to copy a file over and then open it in any other phone app you choose. Second, when WifiTransfer is enabled, it will automatically show up in your file manager’s *Network* section, meaning that you can copy files from your desktop with just one click,” confirmed Stuart Langridge, a member of the Web Standards Project’s DOM Scripting Task Force.

The new version of digiKam 4.13.0 releases with 30 bug fixes

Developers have released the new maintenance version of digiKam; digiKam Software Collection 4.13.0 is now available for download. digiKam, which happens to be an important image manipulation software for KDE/Qt desktop environments, has apparently been released with many new features and almost 30 bug fixes.

“The digiKam team is proud to announce the release of digiKam Software Collection 4.13.0. This release is the result of another huge bugs triage on KDE Bugzilla, where more than 30 files have been closed,” said the company’s official release.



According to the reports, the development cycle of digiKam will be complete with the release of its last maintenance version, which is to be out by October 2015. It’s also reported that digiKam 5.0.0, which would be introduced in the next release, might be ported to KDE Frameworks 5. Tech enthusiasts can download digiKam Software Collection 4.14.0 from the official site. Meanwhile, GNU/Linux users can have a look at the updates through their distribution’s main software repositories.

Facebook said, “After seeing so much internal use of Hack Codegen for diverse applications, it’s our pleasure to open source this library for the external community to use.”

“It is interesting that we are getting back into automated code generation which was a big deal in a different era. Codegen is a valuable new tool that can help developer productivity in the right settings. The issues for automated code generation have typically been learning the abstraction itself, and then the ability to accommodate changes in the generated code. Newer technologies are constantly innovating around these issues,” it added.

Google, Microsoft, Netflix and Cisco join hands to design an open source video format

Seven technology giants—Amazon, Cisco, Google, Intel, Microsoft, Mozilla and Netflix—have come together to create an open source video format. The new project aims to aid the development of new formats, technologies and codecs. The new open source video format will facilitate viewing of UHD videos and other such media over the Web.

These seven tech giants will combine their expertise, with the aim of meeting online demand for high quality audio, video, images and streaming on various devices across the globe.

According to a blog post made by the alliance, the priorities they have set include:

- Being interoperable and open
- Optimised for the Web
- Scalable to any modern device at any bandwidth
- Designed with a low computational footprint and optimised for hardware
- Capable of consistent, highest-quality, real-time video delivery
- Flexible for both commercial and non-commercial content, including user-generated content

With this project, the companies plan to build an ‘open, royalty-free video codec specification, with binding specifications for media format, content encryption and adaptive streaming to accommodate upcoming new media types.’

Mozilla CTO David Bryant explained in a blog post, “One of the biggest challenges in developing open standards in a field like video codecs is figuring out how to review the patents. The Alliance provides a venue for us to share the legal legwork without having to worry about it being used against us down the road.”

“That distributes the load, allows us to innovate faster and cheaper, and gives everyone more confidence that we are really producing a royalty-free codec,” he added.

The Alliance for Open Media is a project of the Joint Development Foundation, an independent non-profit organisation that provides the corporate and legal infrastructure to enable groups to establish and operate standards and source code development collaborations.

Calligra 2.9.7 released with Kexi and Krita improvements

Calligra 2.9.7 is now available as a free update to all open source office suite users. Supposedly the last maintenance release in the 2.9 series, version 2.9.7 brings a lot of improvements to the Kexi and Krita components. Apart from the fixes, it also comes with various new eye-catching features. The new version of Calligra enables its users to set a default paragraph style. It comes with optimised formatting of table cells and enhanced memory usage.

Additionally, it introduces 14px icon sizes to the toolbox.



“The Calligra team is pleased to announce the release of Calligra Suite, and Calligra Active 2.9.7. It is a recommended update that brings further improvements to the 2.9 series of the applications and underlying development frameworks,” said Jarosław Staniek, maintainer and core developer of the Kexi Project.

The latest version no longer copies table cells while copying text. The tooltips now constitute keyboard shortcuts, and the writing direction button has started working correctly. One can also remove a highlighted table by pressing the ‘Backspace’ button on the keyboard.

Also, overwriting a selection when inserting a variable is possible and the icons on the toolbox can now be configured. Meanwhile, sub-folders have now been made for all presets and the colours are set to black.

Tiny Core Linux 6.4 released with a new ASCII Penguin

Robert Shingledecker, the man behind the Tiny Core project, has announced the availability of Tiny Core Linux 6.4. Being a collection of light Linux distros, Tiny Core Linux enables its users to build a full-featured desktop. The operating system was reportedly in the development phase for one month. The latest version of Tiny Core Linux can now be downloaded from the project’s official site. While the release candidate of the distro was rolled out in early September this year, the developers managed to release the stable version within just a fortnight.

Tiny Core 6.4 brings a new ASCII Penguin in MOTD (Message of the Day). It resolves exit code issues in the tce-load component, and brings an update to tce-load’s *recursive_scan_dep* script to create a complete list of dependencies for all applications.

Final version of GhostBSD 10.1 available for download

Eric Turgeon, the man behind GhostBSD, has announced the release of the final version of the GhostBSD 10.1. operating system. This latest version is now available for download with new updates and fixes. Unlike other operating systems, GhostBSD 10.1 is not Linux based, but a major BSD distribution. The



FreeBSD operating system, which is especially designed for end users, comes with the MATE as well as Xfce desktop environments. It offers users the option of installing the GRUB, BSD bootloaders or no bootloader. Additionally, it comes with the Station Tweak utility and Vim text editor software.

The new version of GhostBSD 10.1 also adds the Qt based OctoPkg graphical user interface utility.

With an aim to address the privacy issues, it enables its users to check the efficiency of their passwords. With the help of the Station Update Manager software, it also allows users to install third-party software and update their GhostBSD operating system.

Along with these features, GhostBSD 10.1 brings fixes to various issues regarding the Intel and Radeon/ATI graphics cards, GPT partitions and other errors relating to the installation. With the elimination of the SpiderOak tool, it also replaces the GDM login manager with PCDM.

“After a year of development, testing and debugging we are pleased to announce the release of GhostBSD 10.1 MATE and XFCE which are available on SourceForge, and Torrents for the amd64 and i386 architectures,” confirmed Turgeon.

Black Lab NEXT 2015.9 Beta available for download

Roberto J. Dohnert, the man behind Black Lab Software, has announced the release of the second Beta build of the upcoming Black Lab Linux 2015.12 OS. Backed by the Ubuntu 14.04 LTS (Trusty Tahr) operating system and its Linux 3.19 kernel packages, the Black Lab NEXT 2015.9 Beta is now available for download with new updates and fixes. Built around the Xfce 4.12 desktop environment, Black Lab NEXT 2015.9 comes with the GCC (GNU Compiler Collection) 5, the LibreOffice 5.0 office suite, the Tracker GUI and GNOME Documents applications. “Today we have released Black Lab NEXT 2015.9, which is our second beta for Black Lab Linux 2015.12. With this release, we fixed many of the major issues that dealt with stability. After the Beta release we will be releasing Black Lab Linux with XFCE and GNOME,” confirmed Dohnert.

The recently released version fixes the HTML5 black screen issue in Mozilla Firefox and updates the boot manager to enable installations on UEFI (Unified Extensible Firmware Interface) systems. All the 32-bit dependency libraries get automatically installed, while installing Skype or Wine. Additionally, it brings major fixes to the wireless connections as well.

Black Lab NEXT 2015.9 Beta ports the new XFWM themes along with some beautiful wallpaper. While the developers are all set to bring out the last Beta build of Black Lab Linux 2015.12 in October, tech enthusiasts can download Black Lab NEXT 2015.9 Beta from the official site. However, the pre-release version might include unresolved issues, which can affect the overall user experience.

GNOME 3.18 to include some major improvements in its core components

Tech enthusiasts are desperately waiting for the last milestone in the development cycle of GNOME 3.18. The GNOME developers too are leaving no stone unturned to make it the best open source desktop environment ever. In the midst of hype, it's being heard that the upcoming release candidate (RC) build of GNOME 3.18 will bring some major changes in the core components.



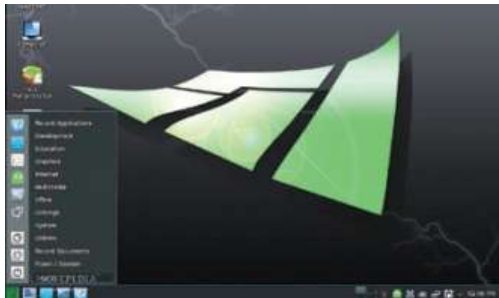
If reports are to be believed, the upcoming version GNOME 3.18 is likely to bring improvements to the Evolution email, calendar and groupware client. Evolution 3.18 RC (3.17.92) will be released with many new features and bug fixes including the addition of an application menu and fixes for a runtime warning. Additionally, it will also come with a feature to avoid vertical scrolling in the filtering rules editor dialogue box.

While most of the changes are still under wraps, reports suggest that the new RC of GNOME 3.18 might come with build changes for Windows platforms and the ECanvas cursor. The EHTML editor component of the Evolution email client is also expected to have major improvements for EHTML editor selection, EHTML editor utils and EHTML editor view. The upcoming version of the desktop environment is likely to add some new languages for its translation menu.

While Evolution 3.18 RC is expected to be distributed as part of the GNOME 3.18 RC build which was scheduled to be released on September 16, the final version of the desktop environment will be available soon.

Manjaro KDE 15.09 features customised Plasma desktop

Manjaro developers have revealed some interesting updates for the recently introduced Manjaro KDE 15.09 RC2. Two RCs of



Manjaro KDE 15.09 are already out and, just a week after the second RC, some of the packages have already received a major overhaul. "It is time for our second release candidate of Manjaro 15.09. Only small adjustments were made. We fixed, for example, our 4.1 kernel series by backporting overlays from 4.2 and Plasma got updated to 5.4.1. Linux316 got updated to 3.16.7.17, such as mesa to 10.6.7 and spl/zfs to 0.6.5. Small changes went into our *manjaro-welcome* package," wrote the developers on the official blog.

Powered by the KDE Plasma 5.4 branch, the new Manjaro KDE 15.09 sports a gorgeous customised Plasma desktop. With the great new look, the system now becomes easily identifiable. It supports the 3.10.87, 3.12.47, 3.13.11.26, 3.14.51, 3.16.7.17, 3.18.21, 3.19.8.6, 4.1.6, and 4.2.0 Linux kernels.

Based on Arch Linux, Manjaro Linux is a distribution which has its own repositories. While it is still in the beta stages, Manjaro claims to be one of the most user-friendly distributions.

**To read more stories about
electronics manufacturing**

ELECTRONICS
ELECTRONICS B2B INDUSTRY
AT THE CLICK OF A BUTTON

Log on to

electronicsb2b.com

www.electronicb2b.com

Log on to www.electronicb2b.com and be in touch with the electronics B2B fraternity 24x7

Open Source Technology: First Love of the Leaders at AskMeBazaar.com!

For some, open source technology is a passion; for some, it is a mission and for start-ups, it is a necessity. **Ekta Mittal**, head, product and technology, **AskMeBazaar.com**, vouches for the fact that open source technologies can truly be the backbone of a tech based start-up like hers. Of course, cost saving makes open source technologies lucrative, but there are other factors too. **Diksha P. Gupta** from **Open Source For You** got in touch with Mittal to explore how start-ups like AskMeBazaar.com benefit from open source technologies. Read on...



Today, after the giant strides taken by open source technologies, you can no longer remain untouched by their magic. Start-ups like AskMeBazaar.com bank on the power of open source for their survival. When asked whether the company uses open source technologies, Ekta Mittal, head, product and technology, AskMeBazaar.com, emphatically states, “Yes, pretty much! We are firm believers in open source technology. Almost over 80 per cent of the technologies that we use in our environment are open source. We are currently working on building an e-commerce platform, in which we are using a lot of open source technologies, including the search platform, the caching technologies, the database, the login mechanism and supporting tools.”

AskMeBazaar is built on the LAMP stack. The tech head elaborates: “The two search platforms we use are Solr and Elastic Search, which are being extensively used these days by most of the companies. Redis is used for caching and Pentaho for reporting. And for detailed analytics and the loyalty programme that we are building, it’s MongoDB. We also work with partners like Unbx, which is built on Elastic Search. Another partner is Vinculum, on which our seller panel is built. The entire e-commerce platform that we have on AskMeBazaar.com is based on OpenCart. So, open source is pretty much everywhere in AskMeBazaar.com.”

And here’s why AskMeBazaar.com uses open source tech...

Apart from factors like having a passion for open source and believing in the philosophy behind it, Mittal uses open source technology at AskMeBazaar.com for certain other reasons too. She says, “There are three key advantages that we get with open source technologies. The first and most important is that one gets control over the code of the software. Community support is another major reason why open source technology is my first choice. Most of the companies, including LinkedIn, Facebook, et al, have started contributing widely to open source technologies. Fluentd, which is a log aggregator, comes from the house of Facebook. I completely believe in the community culture and I want to build a similar culture at AskMeBazaar.com. Third, as the CTO, I have to look at the cost benefits and, with open source, this is one big advantage that any company gets.”

Mittal is not the only one in the company who thinks open source is the best fit. The entire management team backs her on this. She says, “As a company, we are pretty open to open source technologies. When we decide on what technology to use, we evaluate all our options and depending upon what is best with respect to security, we deploy that technology. For example, in the case of analytics tools, we use both Google Analytics and its open source alternative,

Piwik. The latter gives us the liberty to store all the information and Big Data, over which we build a lot of loyalty programmes and event analytics. On the other hand, we use Google Analytics for complete end-to-end analysis of the data. So, I would say we use both technologies, proprietary and open source, depending upon the need.”

Community support is bankable...

While some feel that the open source community support is not something that they can vouch for, Mittal believes that it is pretty bankable. She explains, “If you take Pentaho as an example, it has both an enterprise edition, which is proprietary software, as well as a community edition. So, when you go for the community edition, you can ask for anything including extra bug fixes, various kinds of features from forums like Stack Overflows, etc. They maintain their own JIRAs, where you can request for new updates and also add on to them. These things are super beneficial when you are going with an open source model. Another example that I can cite is that of MongoDB. There are multiple support forums available for it, whereas if you go for an Oracle server or an SQL server, there is minimal support and you have to take help from the proprietary software companies for any kind of development issue.”

Mittal adds, “There have been many cases when the community came to our rescue. Like in the case of Solr, when we were building our entire search over it, we got help from people who have worked on it. There is immediate support available on forums like Stack Overflows, etc. And since there are a lot of people who have already worked on such projects, the support is faster. In case of Unbx, we do a lot of work with them. Although their source code is closed, they are based on Elastic Search, which is open. So there is a lot of exchange that happens between us. This kind of community support really helps a lot.”

There are drawbacks too...

As they say, nothing is perfect and open source technologies are no exception. Mittal has faced some issues as well when she adopted open source technologies. She says, “One of the major factors that one needs to take care of while using any technology, whether open source or proprietary, is that the security standards are maintained and there is no leakage. This goes for open source technologies too. So while you are using OpenCart or Drupal,

you need to do a security review before you take up anything for your project. If you have to deal with data security, then security hardening becomes even more important. This is where proprietary technologies find a place, because they are backed by a company that takes care of these aspects.”

Mittal does her homework before she actually gets on to any technology. She says, “We conduct a lot of security checks before taking up any technology. An example that I can quote is that of OpenCart, where we check for SQL injections, DDoS attack protection, server side validation, et al. OpenCart is a CMS based platform, so a lot of front-end JQuery injection and validations have to be taken care of while implementing it. Also, we ensure that data does not go out of our system because of any leakage. We do the necessary security hardening wherever required, before we deploy any technology.”

Open source: The first love

When it comes to making a tech choice, open source technology is Mittal’s first love. She says, “Before we make a choice, we look

for open source solutions and evaluate them first.

Open source is our priority. In start-ups like ours, open source is actually solving about 80-90 per cent of the software problems. Control over costs really matters for a start-up, which is what we are, currently. We always look for open source. Some open source software is actually much better than the enterprise editions available.

“When we have to make a choice, we definitely look at all the possible available options. That is very important to make an informed choice. Like in the case of the reporting tool that we chose, Pentaho, we did evaluate the advantages of the community edition versus the enterprise edition. If we get better support from the enterprise edition, we will choose that over the community edition, and vice versa,” she adds.

When asked if finding talent in open source software is a challenge, Mittal explains her recruitment strategy. She says, “I won’t deny that we face difficulty in getting talent when we look for in-depth knowledge of open source subjects. But we don’t look for technical brilliance all the time. We look for people who are deeply interested in, passionate and knowledgeable about the basic structure of technologies. If they have the right adaptive attitude and knowledge of basic structures, they can move to any technology, whether it is open or closed source.” **END** 



THE COMPLETE MAGAZINE ON OPEN SOURCE

OpenSource
ForYou

Your favourite Magazine on
Open Source is now on the Web, too.

OpenSourceForU.com

Follow us on Twitter@LinuxForYou

Things to Look at While Choosing an IaaS Provider

This guide will serve as a checklist when choosing the best IaaS provider for your organisation, from amongst the host of options available.



According to Gartner's latest forecast, global spending on IaaS is expected to reach almost US\$ 16.5 billion in 2015, an increase of 32.8 per cent from 2014. The compound annual growth rate (CAGR) from 2014 to 2019 is forecast to be 29.1 per cent. Sunil Gupta, executive director and president, Netmagic Solutions, shares, "Investments in Infrastructure as a Service (IaaS) solutions will be fuelled by the IT Infrastructure expansion happening across Indian enterprises within the service sector, in verticals such as banking, insurance and telecom, as well as within the government segment. IaaS will see a lot of traction, both from large to mid-market enterprises that want to replace their outdated infrastructure, and from small and medium businesses (firms with less than 999 employees). They will be attracted by the promise of developing best-of-breed infrastructure at a low acquisition and maintenance cost. In the near future, usage of data centre and co-location solutions is expected to percolate down to medium and small businesses as well.

Moreover, the government of India's aggressive digitisation plans, as well as the need for documentation of records and regulatory compliance measures, will act as major drivers for the adoption of data centre and cloud services."

With the market growing at a rapid rate, the number of IaaS providers is also growing in India. In fact, it is becoming increasingly difficult for IT decision-makers to choose what is best for their kind of set-up. This guide will serve as a checklist for choosing the best IaaS provider from amongst the host of options available.

Know your needs

Jason Hoffman, head of cloud technology, product area cloud, business unit - Cloud and IP, Ericsson, says, "This is the most

fundamental and obvious step to be taken at the beginning of the process. This step, if well executed, forms the basis for the entire selection. One needs to know the requirements to understand whether IaaS is actually required or building one's own infrastructure is a better option. The IT decision-makers need to understand the reasons for choosing an IaaS provider rather than building their own infrastructure. They need to evaluate the benefits they will get with IaaS. Once they know that, decision making will be much easier."

In addition, IT decision-makers must clearly articulate their expectations from an IaaS provider. They need to make it clear whether they need IaaS for extra computing power or for additional storage, and which of these is the priority.

Robust infrastructure and service consistency

Signing on an IaaS provider means you are outsourcing an important part of your IT infrastructure. Therefore, it is advisable that you choose a fully managed IaaS cloud solution that runs in an enterprise-class data centre. This would mean that it is more secure, and can be relied on to support and deliver hosted software solutions. Anil Chandilya, head - data center operations, "Service consistency is the second important factor, particularly because of the way the hosting and cloud industry has grown and continues to grow, with many providers coming in. Most of them are purely infrastructure providers. Though they invest heavily and build a large infrastructure, most of them do not have any technical support background. Building infrastructure and providing it on lease is just one aspect of the hosting business; however, providing technical support and application uptime is the most important aspect of this industry. So while choosing

a solution provider, it is important to check if the firm has consistently provided technical support services and delivered application level uptime.”

Payment model

Mr Chandilya, shares, “There are two hosting models. One is to pay per allocation, and the second is to pay based on consumption. The first option means paying for whatever is allocated to you, irrespective of whether you use it or not. In the pay for what is consumed model, you pay for whatever you use. An example of the first model is the reality market. If you lease some property, you have to pay for it irrespective of whether you used it or not. An example of paying for what you consume is electricity bills, when you pay for only the amount of power that you use. In our industry, there are providers who invoice for allocation like co-location services and cloud infrastructure. While selecting a provider, it is very important that you pay only for resources you use, like in a utility model.”

Executive dashboard

This is the fourth important aspect to consider when choosing a service provider. Says Mr Chandilya, “It is really important to monitor your services availability. Ultimately, the goal of hosting is to get better application uptime to serve end customers consistently and generate revenue. So it becomes vital to monitor service uptime. How do you know that you are achieving the ultimate goal of hosting? To get this information, it is very important to get an executive dashboard for monitoring, from the service provider. Every role in an

organisation requires a different level of monitoring. Some hosting providers offer a monitoring interface. If the firm permits monitoring of hardware parameters like RAM, CPU, disk and bandwidth usage, this actually helps IT administrators more than it helps CIOs or CTOs, who require end-to-end monitoring of application availability. The latter need a dashboard to monitor if applications are working fine and delivering what is expected. So an executive dashboard is an important feature that customers should ask for.”

Service level agreements (SLAs) and penalties are yet another vital criteria. While choosing any provider, customers should check if they are ready to provide SLAs to underscore their commitments and if they are ready to accept penalties, if the level of service committed to is not delivered. There are multiple factors in an SLA. Sometimes, even though SLAs are in place, the clauses in them are not relevant to the expected service. As solution providers, hosting companies should also help customers to develop the right SLA and deliver based on its terms. For customers, there should be mechanisms to monitor and audit whether services are being provided as promised in an SLA.

Data segregation and recovery

Mitesh Agarwal, VP – sales consulting and CTO, Oracle India, shares, “In the case of public cloud environments, it is critical to make sure hosting providers can guarantee complete data segregation for secure multi-tenancy. Enterprises must also make sure their hosting provider has the ability to do a complete restoration in the event of a disaster.”

A few companies that offer IaaS products



Oracle Infrastructure as a Service (IaaS) Private Cloud: Oracle’s IaaS Private Cloud delivers Oracle Engineered Systems hardware and support for a monthly fee, with no upfront capital expenditure. It combines the security and control of on-premise systems with unique features of cloud computing, including ‘capacity on demand’, which enables businesses to access and pay for peak CPU capacity only when needed.



ESDS’ eNlight Managed Cloud: Since every business is unique in its own way, its cloud solution also needs to be unique. Hence, the firm offers specially engineered intelligent cloud computing architecture for businesses. ESDS’ eNlight claims to give its users the right cloud infrastructure, perfectly pooled resources and dependable technical support to run business enterprises dynamically.



Netmagic Solutions: Netmagic Solutions says it has the depth and breadth of expertise to professionally manage critical IT infrastructure. With its IaaS products, India based enterprises can enjoy immense financial benefits such as cost reductions of between 20-30 per cent of IT operations, decoupling IT costs from people but linking them directly to infrastructure, as well as bringing about some predictability to IT costs.



Note: Details regarding some of the products mentioned in this article have been taken from the Internet.

By: Diksha P. Gupta

The author is senior assistant editor at EFY.

CODE SPORT



In this month's column, we feature a set of computer science interview questions.

Last month, we had discussed a set of questions on operating systems. Let's continue our discussion of computer science questions, while focusing on data management and cloud computing.

1. We are all familiar with transactional databases for storing structured data. However, you are asked to design a data management solution, where the data store needs to store the large documents. The documents are answer sheets of students in a school record management system. Each student is uniquely identified by a student ID, and the data store needs to store all the answer sheets of each student, for all the subjects throughout an academic year, with the ability to store the past three years' records as well. What kind of database would you use? Can you explain the database schema that you would employ?
2. Spatial applications are getting developed in large numbers these days. Consider a taxi fleet management company, where the trajectories of the taxis need to be stored in a database. Spatial applications should be able to query the database for a specific trajectory or segments of a trajectory. It should be possible to answer queries of the form, where given a set of location points, it should be possible to retrieve all the taxis whose trajectories overlap the location points. What kind of database would you use for storing trajectory data? For speeding up query performance, what kind of indexing would you perform for the trajectory data?
3. Assume that you are asked to design the medical data management solution for a big hospital. Medical data generated in the ICU is among the most critical in a hospital, where different sensors generate physiological data also known as vitals. This is a time series data where various physiological signals like heart rate, blood pressure and oxygen level are recorded at periodic time intervals of, say, one minute - for each patient in the ICU. What kind of database would you use to store time series data?
4. Assume that you are asked to design the data management solution of a small scale social network that is becoming popular. The social network has around 100,000 members and is expected to grow by 10X over the next one year. The members are identified by their unique member IDs. Members can have other members as friends. Each member has a unique profile along with a message history, which includes the messages he/she has sent to other members of the network. What kind of database would you use to represent the members of the social network and their friends?
5. Assume that you are asked to redesign the social network site mentioned in Problem 4 as a dating site. The new constraint on the friendship relationship among the members is that they can only become a friend with a member of the opposite sex. How would you implement the dating site's social network? One of the frequent queries that needs to be optimised for the dating site is: Given two members (X,Y) where X and Y are of the opposite sex, do they have any common connections who can introduce them to each other? How would you design your data management solution such that you can speed up this query?
6. Most of you would be familiar with Dockers (<https://www.docker.com/>), which is an open source project built on top of Linux containers. It allows applications to be isolated from each other. Can you explain how Dockers differs in providing isolation to applications compared to virtual machines?
7. You are given a 16 core, 32GB RAM physical machine running Linux. You need to host a Web

server and an application server on this physical server. You need to make a choice between deploying separate virtual machines for the Web server and the app server, and hosting them in two different Dockers containers. What would you choose and can you explain the rationale behind your choice?

8. In Problem 7, you are now told that you don't need to host an application server, but you need to host as many Web servers as possible on the physical machine. Would you now use a virtual machine to host each Web server instance or would you use a Dockers container for each Web server? Can you explain the reason for your choice?
9. Most of you would be familiar with cloud computing services such as Amazon EC2. Amazon also provides cloud storage through Amazon Simple Storage, popularly known as Amazon S3. This object based cloud storage service can be accessed via HTTP services, and your files can be stored as objects in it. What are the advantages of an object based storage service as opposed to a file based storage service?
10. Consider Question 9 and imagine you are running a database server instance on Amazon EC2. How would you host your application data on the cloud? If you are asked to choose between object based cloud storage and block based cloud storage, what would you pick? Can you provide the reasons for your choice?
11. Assume that you have been asked to move your on-premise enterprise payroll application onto the cloud. You have decided on the cloud compute and storage vendor. Your application has 20TB of data, which needs to be hosted onto the cloud. How would you ensure the movement of this data from your on-premise storage to the cloud? What are the issues and precautions that you need to take?
12. Consider that you have stored your critical enterprise data on cloud storage. How do you protect against any data loss due to a failure of the cloud storage system?
13. We are all familiar with the use of RAID in enterprise storage for redundancy so as to avoid data loss. There are two ways of preventing data loss in enterprise storage. One is to have multiple copies of the data. The other is to store the data in erasure coded format so that in the event of a disk failure, the data can be reconstructed from the erasure coded format. What are the pros and cons of these two choices? For cloud storage, would you opt for storing the data in erasure coded format or by keeping multiple copies of the data? Explain the reasons for your choice.
14. Your organisation has been asked to evaluate how using cloud compute and storage compares with using on-premise compute and storage for its IT needs. Which of your enterprise applications would you consider moving to the cloud and why?
15. Assume that you have moved your organisation's payroll data onto the cloud. However, your organisation has four offices and all of them could be updating the payroll database at the same time. How does the cloud storage system synchronise these multiple updates from different geographical locations onto the same data object? While traditional database or file locking would take care of this issue in on-premise data storage, can you use the same mechanism to ensure synchronised, ordered updates from multiple geographical locations onto the same data object? If yes, explain how? If not, explain why traditional file or database locking mechanisms would not serve the purpose?
16. Your enterprise applications have widely varying compute demands, over a period of time. For instance, during month-ends, there is excessive demand for compute to run batch jobs. Similarly, for a couple of hours each night, there is a spike in compute resource usage, again due to batch jobs. How would you handle such dynamic resource demand variations if you need to move your enterprise application onto the cloud?
17. Assume that data redundancy on cloud storage is achieved by means of maintaining three copies of each piece of data. When a user-initiated data change occurs, how are the three copies updated? Would the user-initiated write return success to the user only after all the three copies are updated? Or would it return success right after updating any one of the copies successfully? What are the pros and cons of these two choices?
18. Assume that you have bought cloud compute and cloud storage and have moved your enterprise application onto the cloud. Consider two scenarios—one in which the same vendor provides both compute and storage on the cloud, and another in which the cloud compute and cloud storage come from two different vendors. In each of these scenarios, how do you ensure the locality of data for your cloud hosted application?

If you have any favourite programming questions or software topics that you would like to discuss on this forum, please send them to me, along with your solutions and feedback, at sandyasm_AT_yahoo_DOT_com. Till we meet again next month, happy programming! 

By: Sandya Mannarswamy

The author is an expert in systems software and is currently working as a research scientist in Xerox India Research Centre. Her interests include compilers, programming languages, file systems and natural language processing. If you are preparing for systems software interviews, you may find it useful to visit Sandya's LinkedIn group: *Computer Science Interview Training India* at [http://www.linkedin.com/groups?home=HYPERLINK "http://www.linkedin.com/groups?home=&gid=2339182"&HYPERLINK "http://www.linkedin.com/groups?home=&gid=2339182"](http://www.linkedin.com/groups?home=HYPERLINK%20http://www.linkedin.com/groups?home=&gid=2339182)

“From a personal preference perspective, I would choose an open source solution, any day”

What matters most to a CTO, you may wonder. Well, the answer is obvious—a good solution. Even better would be a good ‘cost-friendly’ solution. This is why open source technologies have been widely adopted by **Anoop Menon, CTO, redBus.in**. Menon is a die-hard fan of open source technologies and promotes these unabashedly across the firm. In an exclusive tête-à-tête with **Diksha P. Gupta** from **Open Source For You**, he shared redBus’ journey with open source. Excerpts:



Q Can you tell us something about the b2b platforms you have?

We have two b2b platforms. One is focused on the bus operators, that is, the businesses. We call it ‘redBus for Businesses’. The early name for this was BOGDS. This is basically the ticketing and inventory management system for the bus operators. They use our system to configure, manage their inventory and find out the route on which the bus travels. They can also use the platform to provide APIs to other partners. The second product we have is called Seat Seller. This is an agent business platform. If you were to travel, say, from Delhi to Jaipur, you can buy bus tickets in three ways. First, you can buy them online. Second, you could go to the bus stand, get into the bus and hope to get a seat, or, third, you can go a day early and buy a ticket at one of the agent shops. These are called travel agents or ticketing agents. We have a product called Seat Seller, which these agents use to book seats for customers who reach out to them for buying tickets. Those tickets are also bought off the redBus platform.

Q What technologies are these platforms based on?

Both BOGDS and Seat Seller use Java at the back-end. They use GWT for the framework and the front-end aspects. They also use the MySQL database and a bit of Redis for caching. This is the broad technology deployment structure at redBus.

Q What kind of open source technology deployments do you have at redBus.in?

Purely from the open source perspective, we use a lot of Java, which is a classic open source language, for both the platforms in our b2b business. Apart from that, we use a lot of open source NoSQL technology, which includes names like Redis, Cassandra and Neo4J graph database, the last of which we use when we want to bring out certain things quickly. We also keep experimenting with a lot of systems around caching and serving. We have played with technologies like MongoDB and Redis Cache. For queueing mechanisms, we use RabbitMQ, which are memory queues that allow us to process the data in a particular fashion. However, there is a small part of our business that still runs on proprietary technology, primarily around .NET.

Q redBus uses a lot of open source technologies. What are the advantages that you derive by using these?

Just to clarify, redBus started off by building everything on .NET a long time back. However, during the course of the last three to four years, we have seen large adoption of open source technologies. Primarily, open source gives us the flexibility to do things fast. If I were to use a particular proprietary technology, I would have to either build every component on my own, or purchase some licensed software. In the first case, I would require a lot of manpower, time and resources. I too was supposed to buy some licensed software, but the cost is usually exorbitant. This is one big advantage that open source brings to the table for redBus.

So, if I want to deliver something new, whether it is a new piece of technology or a process, open source allows me to do it really fast. If a particular process would otherwise take weeks or

months to complete, my team can do it within much less time, thanks to open source technologies. The other advantage is that the support for this technology is really high. Let's say we are using a certain technology and get stuck at some point. There are enough people in the community willing to guide us—those who have already gone through the same problem and have a solution ready. This cuts down a lot on the time we would have spent to get to the bottom of the problem. So community support is the second major advantage that we get with open source technologies.

While open source software is well supported by the community, there are no restrictions on how you use it. You can always take out a piece of open source software, modify it to suit your needs, and use it the way you want. In case of proprietary software, you cannot touch anything because you don't have the source code in most cases. If one requires any changes in the proprietary software, one has to place a request and wait for the approval to arrive. So, there is an issue with proprietary software vis a vis open source technologies.

Q Does talent continue to be an issue when it comes to working on open source technologies?

Not really. There are two aspects to getting the right talent. One is, getting talent to work on specific technologies. This is difficult, unless it is something as popular as Java. However, having said that, there is enough material available on open source. So, as long as I am able to find an engineer who has the ability to grasp such technologies quickly by reading the material available, and is ready to experiment and ramp up quickly, specially in open source skills, I would be happy to hire him or her. For example, if we take the graph database, Neo4J, there is enough content and study material available for it. Just type 'Neo4J' in any search engine and you will probably get hundreds and thousands of pages. So the risk of not finding a person who has worked on a particular platform reduces significantly.

Q Do you get the right kind of talent all the time or do you have to train people in-house?

When we look for talent, we do not necessarily look for talent on a particular technology unless it is something like .NET. What we look for is people who have the ability to ramp up quickly and who have the passion to be able to deliver fast. That is my primary concern when I hire someone. As an organisation, we have a large learning and development (L&D) team, which comes under the recruitment function. We have regular training sessions within the organisation. So, getting someone from within the company to work on a certain technology is easier than hiring someone from outside. However, our core functions are based on Java and getting Java professionals nowadays is not that difficult.

Q What are the challenges you face with open source technologies?

The primary challenge with open source technologies is not being entirely certain about support. What if the support for a certain technology is withdrawn? A few years back there was a framework

for PHP. We started using it, and suddenly we heard that support for that framework had been stopped. Instead, they had started advocating Zend. So we had to give up on that technology and go forward with something more stable. This is one risk that comes with open source technologies.

The other thing is, while you can use open source, there is no accountability for it. If you are using a technology which is pretty new to the system and a problem occurs, then there is nobody to account for it. Let's say there is a bug in the system or a loophole, which may come out only under specific use cases. We have to work to patch that, else it can affect our production environment badly. In case of proprietary systems, there is some accountability, for sure.

While security is very high in open source because a lot of people work on it and usually the bugs are located early, there is also a risk that a lot of people know the technology, inside out. If they know what systems I am using, they can exploit a vulnerability that I don't know about.

Q When did your trust with open source technology begin?

I have officially been a Java guy throughout. In my previous organisation, which was in the telecom domain, I used to write software in Java. Then I moved to Ibibo, where most of the software was being built on Java or PHP. In redBus, when I came in, a part of the system was on .NET and the remaining on Java. Some of these were moved completely onto open source. Our inventory platform is completely on open source right now, while our consumer-facing website, redBus.in, is still on .NET. It's an older system and we would like to go slow with that. Also, we have enough people to manage that system. So, we don't necessarily need to go and change that to open source unless there is a requirement for it.

Q So, if you had to develop an application afresh, today, would open source technology be your first option?

I prefer going the open source way, personally. My recommendation is always to look for a solution that is available on an open source platform, and to check the level of support being offered there. If there is something which is apt and is available in open source, we prefer using that. Apart from getting a lot of community support and other such benefits, there are definitely a lot of cost benefits that an organisation can enjoy, which is really important for us. There are two kinds of costs involved—the development cost and the deployment cost. If I use proprietary software and require some proprietary tools for development, there is a cost involved in that. If there is a licence attached to it, that is an additional recurring cost for running the application. From a cost benefit perspective, I would like to go for open source.

Q What matters to you more—a solution or an open source solution?

I think a good solution matters to me the most. The solution should be scalable. Whether it is built on proprietary technology or open source technology doesn't really matter to me at the CTO level. But, from a personal preference perspective, I would choose an open source solution, any day. As I said, there are a lot of cost benefits that come with open source technologies. **END** 

Malware Analysis Using REMnux

In this second article of the series, we focus on eliminating XOR patterns. Malware tends to employ XOR patterns for self-encryption/decryption because they obfuscate malicious code which, in turn, helps malware avoid detection.



This article gives you an overview of malware code obfuscation. Nowadays, malware tends to remain hidden during infection and operation, avoiding detection and analysis by security tools. Generally, code obfuscation is malware's best friend. It provides high persistence to the malware samples. Code obfuscation is a technique that uses binary and textual data in an unreadable format that cannot be reverse engineered.

A lot of malware have the capability of self-encryption and high persistence. Most malware use the XOR self-encryption/decryption routine. XOR patterns for obfuscating the malicious code are very similar in working patterns:

- i) Out of a possible 255 keys ranging from 0–255, the attacker picks a 1 byte key
 - ii) The attacker obfuscates the code by encoding it with the 1 byte key value, iteratively
 - iii) The attacker can pick the longer key size to obfuscate the code for better persistence
 - iv) The attacker protects the code (encoded string) by iteratively XORing each byte with the key
- XOR is basically very simple, symmetric and easily

reversible with a single functionality for both encryption and decryption. Encrypted malware is hard to detect by the traditional security solutions like anti-virus software, intrusion detection systems and intrusion prevention systems. The only pattern that can be recognised is the XOR pattern used for the encryption/decryption routine. Here, REMnux is used to analyse the malware samples for possible XOR patterns.

NoMoreXor

NoMoreXor is a command line utility used to analyse and guess the 256 byte XOR key using the frequency analysis technique. The utility is bundled in REMnux and is freely available on Git at <https://github.com/hiddenillusion/NoMoreXor>

The syntax is:

```
NoMoreXor.py -a -o [outputfile.extension] [malwaresample.extension]
```

Here's an example:

```
NoMoreXor.py -a -o malware.hex malware.exe
```



```

remnux@remnux:~$ NoMoreXOR.py -a -o malware.hex malware.exe
[+] Attempting auto analysis
[+] HEXing.....: malware.exe
[+] Saving as.....: malware.hex
[+] Attempting to guess the XOR key of....: malware.hex
[+] Size of content.....: 27076210
[+] Total pages (1024k).....: 26441
[+] Total contiguous 512 chunks.....: 52883
[+] Top (5) overall chars
=====
Occurrences | Character(s)
-----
88441 = 0x00
71989 = 0xff
61607 = 0xfe
61067 = 0xdf
60907 = 0xf8

[+] Total number of unique 512 chunks.....: 52877
[+] Top (5) 512 char sequences after cleanup
=====

```

Figure 1: NoMoreXor

```

remnux@remnux:~$ NoMoreXOR.py -a -o malware.hex malware.exe
[+] Attempting auto analysis
[+] HEXing.....: malware.exe
[+] Saving as.....: malware.hex
[+] Attempting to guess the XOR key of....: malware.hex
[+] Size of content.....: 27076210
[+] Total pages (1024k).....: 26441
[+] Total contiguous 512 chunks.....: 52883
[+] Top (5) overall chars
=====
Occurrences | Character(s)
-----
88441 = 0x00
71989 = 0xff
61607 = 0xfe
61067 = 0xdf
60907 = 0xf8

[+] Total number of unique 512 chunks.....: 52877
[+] Top (5) 512 char sequences after cleanup
=====

```

Figure 2: NoMoreXor for extracting possible XOR keys in the sample 'malware.exe'

Balbuzard

Balbuzard is a malware analysis tool that is used to extract patterns from malicious files and to crack obfuscated code using XOR. It is also used to extract strings/patterns and embedded files.

The syntax is:

`Balbuzard.py [option] [filename.extension]`

Here's an example:

`Balbuzard.py --csv malware.csv malware.exe`

Bbcrack

Bbcrack is a supplementary tool used to brute-force XOR keys based on patterns of interest.

The syntax is:

`bbcrack.py [option] [filename.extension]`

An example is:

`bbcrack.py -l 2 malware.exe`

Bbharvest

Bbharvest is also a brute force tool used to extract XOR keys that are specifically targeted against single

```

remnux@remnux:~$ balbuzard.py --csv malware.csv malware.exe
Loading plugin from bbz_sample_plugin.py
Loading yara plugin from clamsrch/clamsrch.yara
Loading yara plugin from malwarecookbook/packer.yara
Loading yara plugin from malwarecookbook/capabilities.yara
Loading yara plugin from malwarecookbook/magic.yara
Loading yara plugin from AlienVaultLabs/urasy_skypedat.yara
Loading yara plugin from AlienVaultLabs/mask.yara
Loading yara plugin from AlienVaultLabs/leverage.yara
Loading yara plugin from AlienVaultLabs/hangover.yara
Loading yara plugin from AlienVaultLabs/APT_NGO_wuacit.yara
Loading yara plugin from AlienVaultLabs/GeorBotBinary.yara
Loading yara plugin from AlienVaultLabs/kins.yara
Loading yara plugin from AlienVaultLabs/ap1.yara
Loading yara plugin from AlienVaultLabs/fpu.yara
Writing output to CSV file: malware.csv
Opening file malware.exe
=====

```

Figure 3: Balbuzard

```

remnux@remnux:~$ bbcrack.py -l 2 malware.exe
Loading transform plugin from /opt/remnux-balbuzard/plugins/trans_
sample_plugin.py
Opening file malware.exe
STAGE 1: quickly counting simple patterns for all transforms
Best score so far: Identity, stage 1 score=27875360
Transform xorA7: stage 1 score=178776

```

Figure 4: Bbcrack

```

remnux@remnux:~$ bbharvest.py -t list malware.exe
Loading transform plugin from /opt/remnux-balbuzard/plugi
ns/trans_sample_plugin.py
Available transforms - Level 1:
- identity: Identity Transformation, no change to data. P
arameters: none.
- xor: XOR with 8 bits static key A. Parameters: A (1-FF)
- add: ADD with 8 bits static key A. Parameters: A (1-FF)
- rol: ROL - rotate A bits left. Parameters: A (1-7).
- xor_rol: XOR with static 8 bits key A, then rotate B bi
ts left. Parameters: A (1-FF), B (1-7).
- add_rol: ADD with static 8 bits key A, then rotate B bi
ts left. Parameters: A (1-FF), B (1-7).
- rol_add: rotate A bits left, then ADD with static 8 bi
ts key B. Parameters: A (1-7), B (1-FF).
Level 2:

```

Figure 5: Bbharvest

obfuscated strings/patterns in malware.

The syntax is:

`Bbharvest [option] [filename.extension]`

Here's an example:

`bbharvest.py -l 2 malware.exe`

Bbtrans

Bbtrans is used to apply transforms over the malware samples.

The syntax is:

`Bbtrans.py [option] [filename.extension]`

And an example is:

Continued to page 32...



Scripting Mail with **SHELL AND PERL**

This is a tutorial aimed at systems and middleware administrators who are new to mailing techniques but know how to use VI, VIM or any other editor. This article will also interest experts and others interested in an easier way of mailing by using a basic system and without having to install extra packages or modules.

Shell and Perl have both been used for varying purposes in the field of systems administration. However, one cannot completely replace the other. So let's look at how to send emails using Shell and Perl on UNIX/GNU Linux platforms. This is very useful while automating alerting or monitoring systems. If you know emailing in Shell, you will find it very easy to translate it in Perl. Both have almost the same lines of code in this article. What we achieve here using Shell and Perl should be equally possible with Ruby, Python or any scripting language that can make use of the existing mailing facilities in UNIX/GNU Linux.

For Perl, we will not use any mailing modules, though that is the most preferred method. This is just to make users aware that Perl can work well even without modules.

Prerequisites

Prior to exploring Shell and Perl, let's just go through the following checklist:

- i) For UNIX or GNU Linux systems, you need Fedora or Ubuntu. I am using Fedora for our examples. So my install pattern will prefer *yum* to *dpkg/apt*.
- ii) You need the Internet to reach the software repository to install packages. Though I said we wouldn't be installing packages, this is just for our test environment. We cannot practice on a production environment, so it is necessary that we have a small server environment installed to see the mailing in action.
- iii) Regarding the mail server, most companies that have Red Hat or Fedora Linux should have Sendmail configured, by default. Some might even have Postfix but the *sendmail* command is very common. The local *sendmail* command will take care of passing the mail onto the local MTA or mail transfer agent (either *sendmail* or *postfix*).
 - a. To keep things really simple, I am using:
 - Sendmail as an MTA
 - Local UNIX services as an MDA (mail delivery agent)
 - The local *mbox* as the mail delivery location
 - The *mail* command as mail client
 - System users for basic authentication and sending/receiving mail
 - b. You need to install Sendmail, as follows:

```
yum install sendmail-cf
service sendmail start
```

- c. VIM or VI Improved is a package that might make your work easier because of the benefit of syntax highlighting. It is very easy to get this package on GNU Linux. On UNIX platforms, we might have to use just the VI editor. (Please check the reference at the end of this article for VIM installations.)
- iv) The following default mail clients can be used:

```
mailx/mail
mutt
OR sendmail -t
```

The above three are the common, basic command line parameters used in UNIX or GNU Linux to send mails. We will be using the last one, *sendmail*, by piping commands to it. This is just to keep things simple and standardised, as we are looking at both UNIX and GNU Linux environments as well as very basic arguments, since all parameters might not be available in all the environments.

The modules normally used in Perl for SMTP are:

```
Net::SMTP
Mail::Sendmail
MIME::Lite (Most widely and commonly used)
```

The procedure

We will avoid importing any Perl modules for actual mailing. Going through these methods, we should be able to learn how a mail is sent out. These are the four types of mails:

1. Clear text email
2. Clear text email with attachment
3. HTML email
4. HTML email with attachment (This is a combination of 2 and 3. Since it is a repetition, we will not actually do this as an example.)

In the examples that follow, do notice the portions that create new lines, in both Shell and Perl. If we miss the right number of new lines, the mail might not produce the desired results.

In Shell, by default, *echo* adds a new line. But in some cases, we need to give one more new line apart from the one Shell already outputs. Perl, by default, does not add a new line; so we have to specifically add the required number of new lines.

In the first case of a clear text email without attachment we might not notice much of a difference. The difference becomes apparent when we have to make more than one MIME addition. Adding files as attachments is one example of going in for MIME additions.

Clear text email without an attachment

On Shell, use the following code:

```
osfyedit@osfymail:~$ mail
Using username "root".
Authenticating with public key "rsa-key-20150405"
Last login: Wed Sep  2 18:34:21 2015 from 192.168.0.13
[root@osfymail ~]# su - osfyedit
(osfyedit@osfymail ~)$ mail
No mail for osfyedit
(osfyedit@osfymail ~)$ mail
Mailroom Mail version 12.5 7/5/10.  Type ? for help.
"/var/spool/mail/osfyedit": 1 message 1 new
>N 1 osfyuser@osfymail.in  Wed Sep  2 19:11  18/690  "Shell Mail without at"
Message 1:
From: osfyuser@osfymail.in  Wed Sep  2 19:11:02 2015
Return-Path: <osfyuser@osfymail.in>
Date: Wed, 2 Sep 2015 19:11:01 -0700
From: osfyuser@osfymail.in
To: osfyedit@osfymail.in
Subject: Shell Mail without attachment
Content-Type: text/plain; charset=us-ascii
Status: R

this is a text for all mailers
Welcome to the world of mailing... Buhahaha!!
$
```

Figure 1: Mail without attachment

```
#!/bin/bash
```

```
subject="Shell Mail without attachment";
from="osfyuser"; #can have complete mailing email id
to="osfyedit"; # These are local mail users
```

```
bodyFile="mailbody.txt";
cat > $bodyFile <<EOF
This is a text for all mailers
Welcome to the world of mailing... Buhahaha!!
EOF
```

```
{
echo "From: $from"
echo "To: $to"
echo "Subject: $subject"
echo -e "Content-Type: text/plain; charset=us-ascii\n"
cat $bodyFile
echo
} | /usr/sbin/sendmail -t
```

- i) As root, create two users, as shown below:
- a. osfyuser

```
useradd osfyuser
```

- b. osfyedit

```
useradd osfyedit
```

- ii) Log in as *osfyuser*. Use the editor of your choice to save the above code as a file. Let us call it *mailer.sh* and make it executable (we're using VIM), as follows:

- a. `chmod u+x mailer.sh`

- iii) Run the executable file, as follows, so that it sends a text mail to the user *osfyedit*:

- a. `./mailer.sh`


```

osfyedit@osfymail:~$ mail
osfyedit@osfymail ~$ mail
Heirloom Mail version 12.5 7/5/10. Type ? for help.
~/var/spool/mail/osfyedit: 1 message 1 new
N 1 osfyuser@osfymail.in Wed Sep 2 19:16 29/895 "Shell Attachment"
Message 1:
From: osfyuser@osfymail.in Wed Sep 2 19:16:39 2015
Return-Path: <osfyuser@osfymail.in>
Date: Wed, 2 Sep 2015 19:16:38 -0700
From: osfyuser@osfymail.in
To: osfyedit@osfymail.in
Subject: Shell Attachment
Content-Type: multipart/mixed; boundary="frontier"
Status: R

Part 1:
Content-Type: text/plain; charset=us-ascii

this is a text for all mailers
welcome to the world of mailing... Buhahaha!!

```

Figure 2: Mail with attachment

- iv) A file gets created so as to represent the body of the mail:

```

[osfyuser@osfymail ~]$ ls -l mailbody.txt
-rw-rw-r-- 1 osfyuser osfyuser 76 Aug 31 19:05 mailbody.txt

```

Open another terminal, log in as *osfyedit* user and type 'mail' to see if there are any mails.

On Perl, after saving the following code as a file in *osfyuser*, with the filename *mailer.pl*, repeat the procedures from iii) to v) given above, and compare the results. They should be exactly the same.

```

#!/usr/local/bin/perl

my ($subject, $from, $to, $fh, $content);
$subject = "Perl Mail without attachment";
$from = "osfyuser";
$to = "osfyedit";

$fh="mailbody.txt";
open(FILE, ">", "$fh") or die "Cannot open $fh: $!";
print FILE "This is a text for all mailers\nWelcome to the
world of mailing... Buhahaha!!";
close(FILE);

open(MAIL, "|/usr/sbin/sendmail -t");
print MAIL "FROM: $from\n";
print MAIL "TO: $to\n";
print MAIL "Subject: $subject\n";
print MAIL "Content-Type: text/plain; charset=us-ascii\n\n";
open(FILE, "<", "$fh") or die "Cannot open $fh: $!";
print MAIL <FILE>;
close(FILE);
print MAIL "\n\n";
close(MAIL);

```

Clear text email with an attachment

Normally, we use uuencode, the universal method of converting a file into understandable UNIX code that is decoded at the server end.

For Shell, use the following code:



Figure 3: Mail listing with browser mail client and IMAP mail service



Figure 4: Mail without attachment read with browser mail client

```

#!/bin/bash
subject="Shell Attachment";
from="osfyuser";
to="osfyedit";
attachment="/home/osfyuser/test.txt"

bodyFile="mailbody.txt";
cat > $bodyFile <<EOF
This is a text for all mailers
Welcome to the world of mailing... Buhahaha!!
EOF

{
echo "From: $from"
echo "To: $to"
echo "Subject: $subject"
echo "Content-Type: multipart/mixed; boundary=\"frontier\"";
echo "--frontier"
echo -e "Content-Type: text/plain; charset=us-ascii\n"
cat $bodyFile
echo
echo "--frontier"
echo -e "Content-Disposition: attachment; filename=`basename
$attachment`"
echo -e "Content-Type: text/plain; name=$attachment\n";
cat $attachment
echo
} | /usr/sbin/sendmail -t

```

This time, we will have an extra line while checking for mail. The attachment, which is in plain text, will turn out as readable text in the destination mail box.

This is a squirrelmail client tool which is a browser based view of the mails sent from *osfyuser* on the command line to user *osfyedit*. I used *Dovecot IMAP/POP* to make the browser connect to a mailbox (see Figure 4).

For Perl, use the following code:

```
#!/bin/perl
```

```
my ($subject, $mach, $from, $to, $attachment, $fh, $content);
$subject = "Test Mail";
$from = "osfyuser";
$to = "osfyedit";
$attachment="/home/osfyuser/test.txt";
```

```
$fh="mailbody.txt";
open(FILE, ">", "$fh") or die "Cannot open $fh: $!";
print FILE "This is a text for all mailers\nWelcome to the
world of mailing... Buhahaha!!";
close(FILE);
```

```
open(MAIL, "|/usr/sbin/sendmail -t");
print MAIL "FROM: $from\n";
print MAIL "TO: $to\n";
print MAIL "Subject: $subject\n";
print MAIL "Content-Type: multipart/mixed;
boundary=frontier\n";
print MAIL "--frontier\n";
print MAIL "Content-Type: text/plain; charset=us-ascii\n\n";
open(FILE, "<", "$fh") or die "Cannot open $fh: $!";
print MAIL <FILE>;
close(FILE);
print MAIL "\n\n";
print MAIL "--frontier\n";
chomp(my $basename=`basename $attachment`);
print MAIL "Content-Disposition: attachment;
filename=$basename\n";
print MAIL "Content-Type: text/plain; name=$attachment\n\n";
open(FILE, "<", "$attachment") or die "Cannot open
$attachment: $!";
print MAIL <FILE>;
print MAIL "\n";
close(FILE);
close(MAIL);
```

For sending *.pdf*, *.zip* or *.doc* or *.xlsx* files, we need to use:

- i) uuencode or base64 for Shell
- ii) base64 and the IO::File module for Perl

The following are only small snippets we can adjust/add to the code provided earlier.

For Shell, the snippet is:

```
echo "Content-Transfer-Encoding: uuencode"
#echo "Content-Transfer-Encoding: base64"
echo -e "Content-Disposition: attachment; filename=`basename
$attachment`"
echo -e "Content-Type: application/octet-stream;
name=$attachment\n";
uuencode $attachment `basename $attachment`
#base64 $attachment
```



Figure 5: Mail with attachment seen with browser mail client

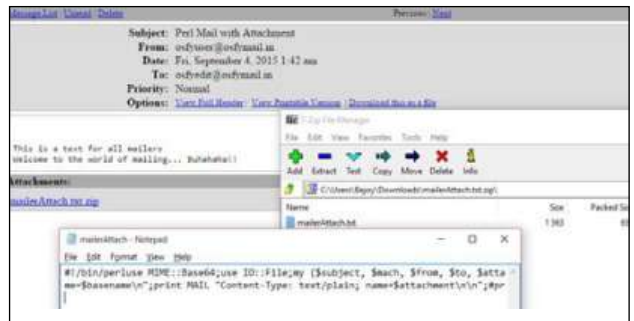


Figure 6: Attachment opened from the browser using Zip utility

For Perl, the snippet is:

```
#print MAIL "Content-Transfer-Encoding: base64\n";
#print MAIL "Content-Transfer-Encoding: uuencode\n";
print MAIL "Content-Disposition: attachment;
filename=$basename\n";
print MAIL "Content-Type: application/octet-stream;
name=$attachment\n\n";
# print MAIL read_file($attachment); # Can be also used for
text/plain attachments
print MAIL encode_base64( read_file($attachment) ); # CTE must
be 64; this must be accompanied by the MIME::Base64 module
which provides this function for encoding. It comes default
on all Perl installs.
#print MAIL `base64 $attachment`; # CTE must be 64; this
procedure is incase you do not want to use the above module.
Using the Shell command ticks.
#print MAIL `uuencode $attachment $basename`; # CTE must
be uuencode; this also uses the Shell command ticks to use
uuencode instead of base64
```

The following function is required for Perl to read the file properly, only if we use the `encode_base64` function from the `MIME::Base64` module. We could also use this function for plain text as in a commented portion in the code earlier.

```
sub read_file
{
    my $filename = shift @_;
    my $attach = new IO::File;
    $attach->open("< $filename")
        or die "Error opening $filename for reading - $!\n";
```

```
$attach->binmode;
local $/;
<$attach>
}
```

Ensure that you use the right command for the right content transfer encoding. Whichever line comes before the *uuencode* or *base64* command, it must have two new lines before it, to process the mail properly. In Shell, by default, *echo* comes with a new line. Adding '-e' option and '\n' would make that two new lines.

The *Content Type* can be figured out using the *file* command in Linux. But in other UNIX platforms, the *file* command doesn't have that feature. Keeping it as 'octet-stream' handles any kind of file. By default, when we attach any file, the normal *mail* command takes the default as octet-stream, except for plain text files.

HTML email

We have so far handled only plain text mails. And if we receive text mails in Outlook or any other mail client, when we try to reply, it (by default) sticks to the non-HTML environment in which it was initially received. Only in HTML format can we highlight or beautify the text if we need to forward a text mail to someone. To do that, we have to manually change the property of the mail from text to HTML. Instead, we can also make the machine send an HTML mail rather than a text mail.

We can also add different kinds of HTML entries, like tables, URLs or any form of coloured mail, even using style sheets. The presentation always matters.

The following code will send HTML mails. I'm not covering HTML with attachments. If you want to add attachments, the earlier attachment methods could be used (plain text or application/octet-stream and encoding standards to be base64 or normal 7-bit).

For Shell, the code is:

```
#!/bin/bash

subject="Bash HTML";
from="osfyuser";
to="osfyedit";
url='http://localhost/css/mail.css';
#css='location/to/css/file' # If we are using a local
stylesheet, then `cat $css` into the body
css='mail.css' # If we are using a local file, then cat $css
into the body
`wget -qO- $url > $css`;
bodyFile="mailbody.html";

cat > $bodyFile <<EOF
<!doctype html public "-//w3c//dtd html 4.0 transitional//
en">
```

```
<html>
<head><style>
`cat $css`
</style></head>

<p>this is test mail<br>

<table id='mail'><tr><th>Month</th><th>Savings</th></tr><tr>
<td>January</td><td>\$100</td></tr>
<tr class='alt'><td>February</td><td>\$500</td></tr></table>
</html>
EOF

{
echo "From: $from"
echo "To: $to"
echo "Subject: $subject"
echo -e "Content-Type: text/html; charset=us-ascii\n";
cat $bodyFile
echo
} | /usr/sbin/sendmail -t
```

For Perl, the code is:

```
#!/usr/local/bin/perl
use LWP::Simple qw(get);

my ($url, @css, $subject, $from, $to, $fh, $content);
$subject = "Perl HTML";
$from="osfyuser";
$to="osfyedit";
$fh="mailbody.html";
$url = 'http://localhost/css/mail.css'; @css = get $url;
#@css=`cat $css`;

my $message=<<"EOF";
<!doctype html public "-//w3c//dtd html 4.0 transitional//
en">
\n<html>\n
\n<head>\n<style>\n
@css
</style></head>\n\n
<p>this is test mail\n\n
<table id='mail'>\n<tr>\n<th>Month</th>\n<th>Savings</th></tr>
<td>January</td>\n<td>\$100</td></tr>\n<tr class='alt'>\n
<td>February</td>\n<td>\$500</td></tr></table>\n
</html>\n
EOF

open(FILE, ">", "$fh") or die "Cannot open $fh: $!";
print FILE $message;
close(FILE);
```



```
open(FILE, "<", "$fh") or die "Cannot open $fh: $!";
open(MAIL, "|/usr/sbin/sendmail -t");
print MAIL "FROM: $from\n";
print MAIL "TO: $to\n";
print MAIL "Subject: $subject\n";
print MAIL "Content-Type: text/html; charset=us-ascii\n";
print MAIL <FILE>;
close(FILE);
close(MAIL);
```

Emailing via PHP is well explained in the following URL for all the above three cases, though it does not include style sheets: http://webcheatsheet.com/php/send_email_text_html_attachment.php



Note: There are two major differences between PHP and Shell/Perl when it comes to effective emailing:

- i) PHP is itself embedded HTML, while in Perl we have to separately embed to make it HTML.
- ii) PHP has a built-in mail function, which does a major chunk of the mailing. We could very well write a Perl module or Shell function file that we could reuse or call in programs. I prefer a Perl module because of the level of intricacies that can be included to make it a perfect function; and then we can use it within a Shell program too.



Figure 7: Mail with HTML contrary to normal text



References

- [1] <https://en.wikipedia.org/wiki/MIME>
- [2] http://www.perlmonks.org/?node_id=675595 à Net::SMTP
- [3] <http://perldoc.perl.org/MIME/Base64.html>
- [4] <http://search.cpan.org/~mivkovic/Mail-Sendmail-0.79/Sendmail.pm>
- [5] <http://search.cpan.org/~rjbs/MIME-Lite-3.030/lib/MIME/Lite.pm>
- [6] <http://www.if-not-true-then-false.com/2012/vi-vim-syntax-highlighting-on-fedora-centos-red-hat-rhel/>
- [7] <http://www.vim.org/download.php>
- [8] <http://perl.plover.com/local.html>

By: Bejoy Abraham Mathews

The author works as a middleware admin at Perl coding expertise, located at Irvine, California. He has earlier authored articles on Puppet and Saltstack. He can be contacted at bejoy.abraham@gmail.com

Continued from page 26...

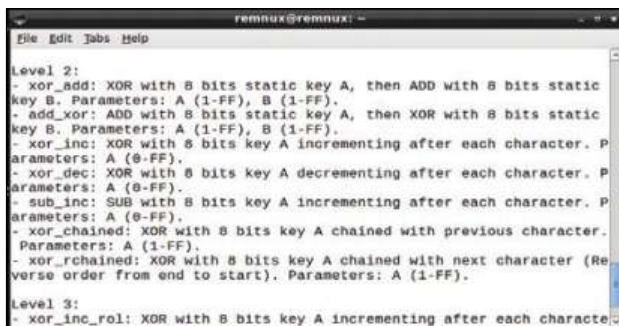


Figure 6: Bbtrans

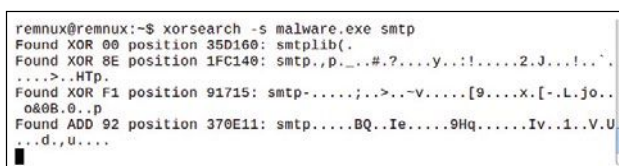


Figure 7: Xorsearch

`bbtrans.py -l 2 malware.exe`

Xorsearch

Xorsearch tools help to find all possible 1 byte key values. However, the particular string that we are looking for is

known and one good value is 'smtp', because some malware samples often send mail to the command and control server.

The syntax is:

`xorsearch [option] [filename.extension] <string>`

An example is:

`xorsearch -s malware.exe smtp`

References

- [1] <https://en.wikipedia.org/wiki/Malware>
- [2] www.threattracksecurity.com/enterprise-security/malware-analysis-sandbox-software.aspx
- [3] <https://zeltser.com/build-malware-analysis-toolkit/>
- [4] <https://isc.sans.edu/forums/diary/Wipe+the+drive+Stealthy+Malware+Persistence+Mechanism+Part+1/15394/>
- [5] <http://digital-forensics.sans.org/blog/2013/05/14/tools-for-examining-xor-obfuscation-for-malware-analysis?reply-to-comment=15387>
- [6] <http://www.decavage.info/python/balbuzard>

By: Sibi Chakkaravarthy Sethuraman

The author holds an M. Tech degree in computer science and engineering, and is currently pursuing a PhD at Anna University. He is with the department of electronics engineering, MIT, Chennai, and can be reached at sb.sibi@gmail.com

Openjudge: An Online Judge for Offline Programming Contests



Do you need an impartial judge for a programming contest? Openjudge is a LAN based program judging implementation in Python. Get a taste of it in this introductory article.

Almost all institutes have a computer science department or an IT department, nowadays. Very often, these departments organise programming contests. While participating in these contests, I became aware of the following things:

- Eighty per cent of the time, the programs are checked manually
 - A lot of people participate
 - People prefer the contests to be online (like codechef.com)
 - Offline events have their own charm and draw large crowds
- The advantages offered by online program judges

are numerous:

- Multiple language support
- Quick checking
- No biases

Openjudge bridges the gap between online judges and offline contests. It offers all the capabilities of an online

judge for an offline system, with very few requirements. One can set up a programming competition with minimal effort spent in developing the infrastructure needed for it. So let us get started.

The set-up

Let us explore the individual components of Openjudge. To install it, we first need some configuration. Openjudge was written with Linux in mind in order to eliminate complex issues arising from supporting multiple operating systems. This article was written using Linux Mint 17. It is known to work on Ubuntu also. The prerequisites are:

1. Linux Mint/Ubuntu
2. Python 3
3. Git
4. Pip and Virtualenv

In order to complete this list of requirements, let's first get

hold of a Linux system. After that, open the terminal (*Ctrl + Alt + T*) and type in the following commands:

```
$ cd ~

$ mkdir judge
$ cd judge

$ sudo apt-get install python3-dev
$ sudo apt-get install python3-virtualenv

$ virtualenv -p python3 env
$ source env/bin/activate
$ pip install openjudge

$ sudo apt-get install git
$ git clone https://github.com/theSage21/judge-interface.git
$ ./setup.sh
```

The first part creates a folder in your home directory called *judge*. The second part installs Python 3 and Python development packages. The third part creates a virtual environment and installs Openjudge in it. Since Openjudge needs an interface to function, the fourth part downloads the default interface and runs the set-up for it.

In the fourth part, make note of the dot in the end of the command. It is important.

While setting up, you will be asked to create a superuser for the competition. The superuser is one who adds or removes questions, languages of submission and participants.

Usage

The *judge* is simple to use. In order to host a programming competition, you must have a LAN. It does not matter if the LAN is wired or wireless. After that, the steps are as follows. In a terminal, type the following commands:

```
$ cd ~/judge
$ ./runserver.sh
```

In another terminal, type:

```
$ cd ~/judge
$ source env/bin/activate
$ openjudge
```

The order is important. The interface must be started first and then the *judge*. That is all. In order to make this competition available to all computers on LAN one must only supply the IP address of the computer on which the *judge* is running. You can find the IP address of the computer by running *ifconfig* in a terminal window. Here is an example from mine.

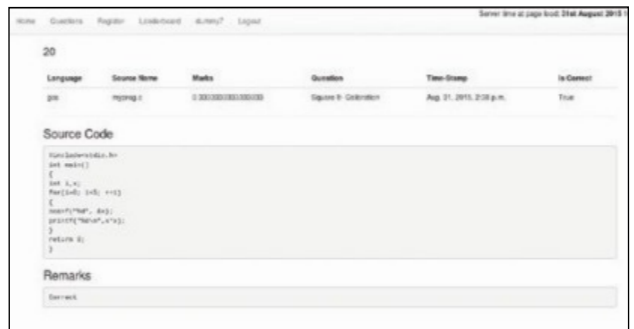


Figure 1: Attempt details



Figure 2: Home page

Rank	Player	Score	LastTime
1	dummy	1.0	1 day, 21 hours
2	dummy1	0.21717171717171717	1 day, 3 hours
3	new037sk	0.20000000000000000	22 hours, 7 minutes
4	dummy2	0.0	1 day, 9 hours
5	dummy3	0.0	1 day, 9 hours
6	dummy4	0.0	1 day, 9 hours
7	dummy5	0.0	1 day, 9 hours
8	dummy6	0.0	1 day, 9 hours
9	dummy7	0.0	1 day, 9 hours
10	adwin	0.0	22 hours, 55 minutes
11	shama	0.0	22 hours, 55 minutes
12	adwin2	0.0	22 hours, 55 minutes
13	hllaga 0	0.0	11 hours, 15 minutes

Figure 3: Leaderboard

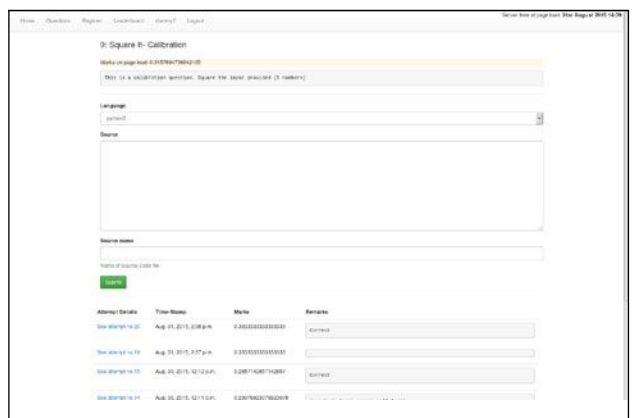


Figure 4: Question view

```
ghost@amnesia $ ifconfig
wlan0 Link encap:Ethernet HWaddr c4:17:fe:b8:ab:f6
       inet addr:192.168.10.116 Bcast:192.168.11.255
       Mask:255.255.254.0
       inet6 addr: fe80::c617:feff:feb8:abf6/64 Scope:Link
```


IT'S TIME YOUR PHOTO APPEARED ON OUR COVER....

Inviting our readers to create cover designs of *Electronics For You* that feature them and their loved ones.

Just download the ALIVE app, scan the EFY cover featured here, and replace the photo on the cover with your own -- and there you are, **on the cover of *Electronics For You*!**



Step 1

Get the free Alive app:

Give a missed call to

18001023324

or visit

www.alivear.com/app

from your mobile phone.



Step 2

Picture scan:

Open the ALIVE app on your phone and scan the EFY cover above using your phone's camera. Replace the photo in this cover with your own.

Step 3

Now, just share this with your friends and family, and they will see you on the cover of South Asia's leading tech magazine!

```
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:1947721 errors:0 dropped:0 overruns:0 frame:0
TX packets:1064025 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2861465984 (2.8 GB)  TX bytes:120199975 (120.1 MB)
```

...

This shows that my wireless network card `wlan0` has an IP address of `192.168.10.116`. If I have set up Openjudge on my computer, anyone on the WLAN can access the competition by typing `192.168.10.116` in their respective browser windows.

Note that since we have created a virtual environment, the `openjudge` command will only run when the environment is active. To do that, navigate to the folder you installed Openjudge in. In this tutorial, it is `cd ~/judge/`. After navigating to the folder, you can activate the environment by typing `source env/bin/activate`.

After this, Openjudge is available for use. Virtual environments let us use different versions of software on the same computer. When a new version of Openjudge comes out, you can test it by just creating a new folder and typing the following code:

```
$ virtualenv -p python3 env
$ source env/bin/activate
```

```
$ pip install openjudge
```

Thus you can use two different versions without having to reinstall them, over and over again.

Features of Openjudge

- 1) The superuser can control the competition via the *Administration* interface, which is available at the URL `http://<IP_ADDRESS>/admin/`.
- 2) Multiple languages can be used in the programming contest. Initially, the *judge* only supports Python2, Python3, C, C++ and Java. Others can be added as per the need.
- 3) Complete details are available in the terminal running Openjudge. The *judge* also creates a *logfile* to keep track of things that have been done. **END** 

References

- [1] Openjudge: <http://theSage21.github.io/openJudge/>
- [2] Interface: <http://theSage21.github.io/judge-interface/>
- [3] Issues may be reported at <https://github.com/theSage21/openJudge/issues/>

By: Arjoonn Sharma

The author is a Python developer who candidly says that he is fresh out of college. You can link up with him via github.com/theSage21 or arjoonn.blogspot.com

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK

We Value Your Feedback

We love to hear from you as **OSFY** consistently strives to make its content **informative, interesting and engaging.**

Please share your feedback/ thoughts/ views via email at osfyedit@efy.in

We welcome your comments/ suggestions and encourage you to send them to:
The Editor, D-87/1, Okhla Industrial Area, Phase 1, New Delhi-110020.

OpenSource
THE COMPLETE MAGAZINE ON OPEN SOURCE **For You**

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK



This article, which features chefs, recipes and knives, isn't about cooking in the kitchen. It heralds a new series of articles in the Dev-ops corner. Subsequent articles in the series will explain how to configure Chef agents in workstation nodes and how to link the agents to the Chef server. We will also cook some recipes for Chef configuration.

Software product development consists of a series of planned activities. During the SDLC (software development life cycle) phase of program development, repetitive actions are required for design refinement, review fixes, build, deployment and regression testing. With the advent of newer programming methodologies and modes of execution like agile and extreme programming, there is a need to automate such tasks to a greater extent.

When we automate various stages in the SDLC cycle, depending on the methodology we adopt for the project or the time line for execution, there is a greater demand for the repeated actions of build, deployment, unit testing and promoting the build to the next stage. These activities occur at multiple stages (development, integration testing, system testing, user acceptance and load testing) and repeatedly (for various phases of releases). Saving the time spent in such

complex and repetitive tasks helps in improving productivity and reducing manual failures.

So, there is a definite need for automating repeated tasks in build preparation, build and packaging, deployment and in unit/regression testing. This process is called continuous delivery. Continuous delivery is the process of automating the build/release process in the life cycle of software configuration and delivery. It differs from regular build/deployment in that it is a process of automatic delivery with less manual intervention and runs automatically during code check-ins.

Chef is a popular automation platform for this process of continuous delivery, as it considers infrastructure as a code and helps in build/deployment regularly; it also integrates automatic regression testing to enable certification of the build for promotion to the next stage. Chef helps the software configuration manager to programmatically handle build, deploy,

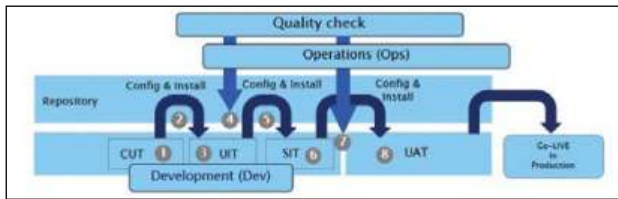


Figure 1: Dev-Ops architecture

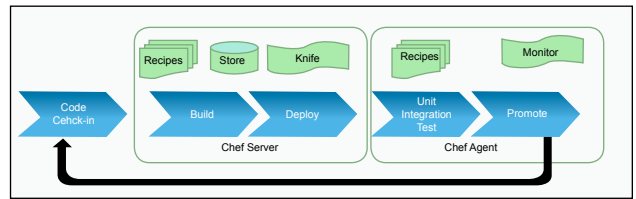


Figure 3: Chef operations in the server and agent

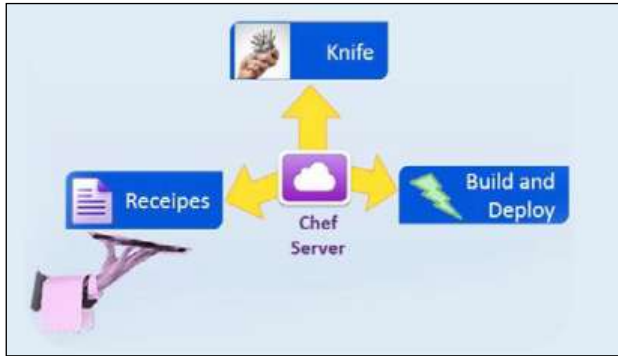


Figure 2: Components in Chef Server

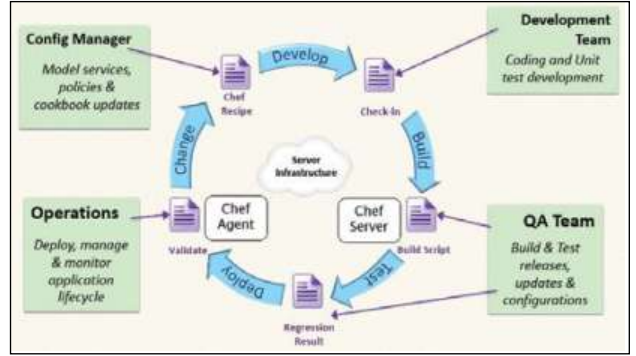


Figure 4: Chef tools

test and to promote code builds into various server zones. This is also technically termed as ‘turning infrastructure into code’.

Later, we plan on writing a series of articles on Dev-Ops. For now, let’s start with an introduction to Chef, its journey so far, and then install and configure Enterprise Chef on RHEL 7, which will help you to create a Chef platform to work further.

The evolution of Chef

Chef was first released in January 2009 in Linux based continuous deployment (CD); it stabilised around 2010 in Linux as well as Windows. Its original name was Marionette, when Adam Jacob created it as a pilot for automating the build/deployment process for one of his clients. The potential of this tool was realised when it began handling automation efficiently in Amazon work servers. That’s when it was converted to a product and named Chef. It is based on scripts called recipes, which run to push components for deployment and do other build/deployment activities. This set of recipes for a unique deployment environment is called a cookbook. The name Chef, therefore, is so apt and the tool used by Chef for such deployment is also called a knife. Chef and its supporting tools are written using the Ruby language.

Chef incorporates in a unified convenient place, all the development, build, deployment and monitoring tools and services needed to move a software product from initialisation and set-up, through development, on to testing and, finally, release.

Chef enables organisations to quickly deploy, manage and scale existing and new enterprise-grade applications on any infrastructure, with no changes to the code or architecture. It helps to maximise application onboarding and automation by externally orchestrating the application deployment and runtime. The Chef cookbook based approach treats

infrastructure as code, enabling you to describe deployment and post-deployment steps for any application through an external blueprint called a recipe, which you can then take from one infrastructure to another, unchanged.

Chef is highly useful in cloud based deployment as well, and simplifies cloud migration by provisioning the compute resources you need, on demand, on the cloud of your choice (public cloud, private cloud, or even hybrid cloud), using its deployment tool to push to the cloud. You can define scaling rules based on any custom metric—and Chef will scale out or scale in as needed. Technically, Chef enables you to install your application with a single shell or REST (Representation and State Transfer) API, with no code changes, and also takes your infrastructure monitoring capabilities to the next level. You can easily monitor your application services for availability and performance, define any custom metrics you want to use, and ensure business continuity at all times.

Chef’s components

Chef Server or Enterprise Chef can be installed on the Windows platform, Mac OS X or on the Linux platform. We have two flavours of Chef available in Linux—the Red Hat Enterprise version and the Ubuntu version.

The Chef client is available in UNIX flavours like FreeBSD, Linux, AIX and Solaris, in addition to Windows and Mac OS. Hence, we can set up Chef nodes in all these machines and integrate with a primary hosted Chef server.

Chef has two flavours of installation, namely, Hosted Chef and Standalone or Enterprise Chef. Let us discuss them briefly.

Hosted Chef runs on a preconfigured cloud environment, where the user has to reserve a space for deployment activities and then upload code for automating build, deployment, testing and to promote the build. All required



Figure 5: VMware configuration for RHEL v7

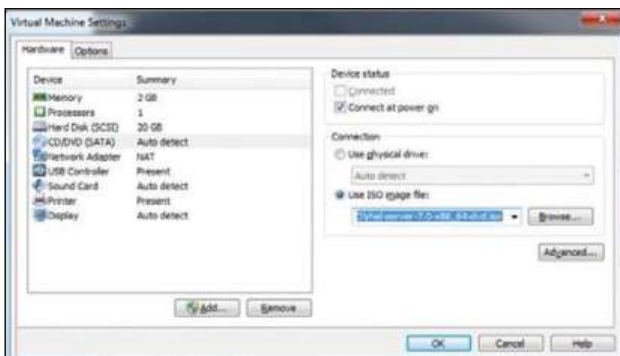


Figure 6: Selecting the ISO—it decides which OS is to be prepared for VM

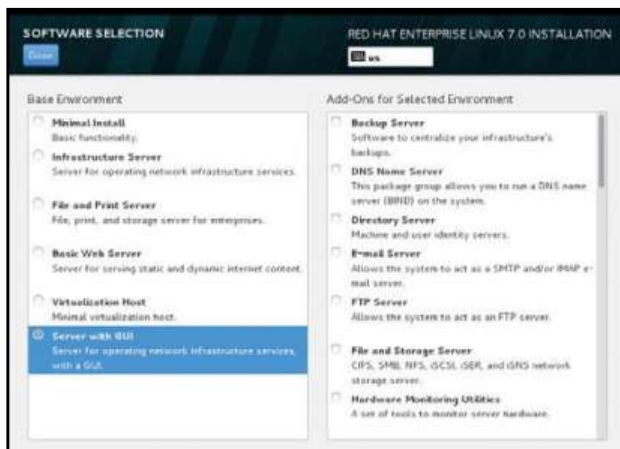


Figure 7: Selected configuration for VM

tools for Chef are pre-installed in the hosted platform, and we need to use the predefined platform for our deployment.

Standalone Chef or Enterprise Chef is an on-premise Chef installation, for which users have to provision infrastructure (servers) for Chef tools, and install/configure and maintain the Chef suite along with application configuration and set-up. In a nutshell, Hosted Chef is easy to kick-off for piloting, provisioning and for quicker setting up, whereas Standalone

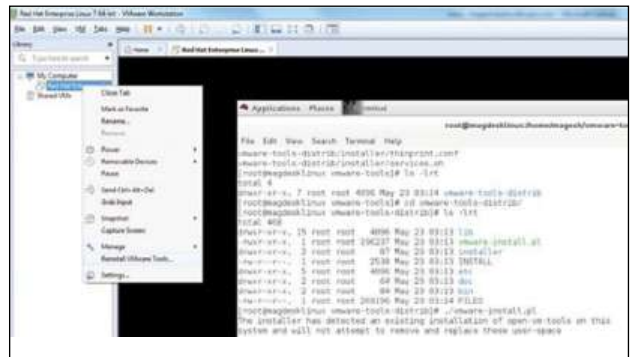


Figure 8: Installing VMware tools

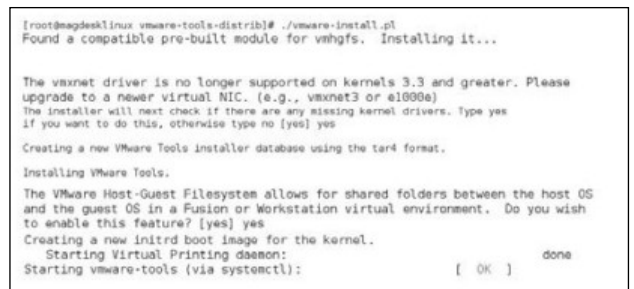


Figure 9: VMware tools installation

Chef is meant for core customisation and secured on-premise set-ups. With the open source Chef platform, Standalone Chef can be customised to a greater extent like including a public key infrastructure (PKI) based authentication mechanism, IAM (identify and access management) tool integration for roles, rights based server handling, etc.

Chef tools

Chef Server: This helps in automating infrastructure and bundles with Webserver, the Ruby script builder, a deployment manager and monitoring tool; it is used more for server handling operations.

Chef Development Kit: This is also called Chef DK and is used for the customisation of Chef Community scripts, cookbooks and deployment actions.

Chef client/Agent: Each node of the connected system in the Chef managed servers should have a Chef client, which acts as a bridge for communicating with Chef Server and the deployment zone.

The Chef installation procedure

Installing and configuring Chef, as well as preparing recipes and cookbooks for production level deployment, requires a lot of analysis, testing and scaling improvements. Ideally, this needs to be done in a sandbox environment, also called a simulated environment, and should be tuned fully before going live in a production environment with multi-node handling.

The key reason here is that we save a lot of time and investment in test deployment, tuning the script

```
[root@magdesklinux Downloads]# sudo rpm -Uvh private-chef-11.3.2-1.el6.x86_64.rpm
warning: private-chef-11.3.2-1.el6.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
Preparing...
Installing...
1:private-chef-11.3.2-1.el6
Thank you for installing Private Chef!
[root@magdesklinux Downloads]#

[root@magdesklinux Downloads]# rpm -Uvh chef-server-core-12.1.2-1.el7.x86_64.rpm
warning: chef-server-core-12.1.2-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
warning: waiting for transaction lock on /var/lib/rpm/rpm.lock
Preparing...
Updating / installing...
1:chef-server-core-12.1.2-1.el7
Cleaning up / removing...
2:private-chef-11.3.2-1.el6
```

Figure 10: Installing Chef Server components

```
[root@magdesklinux Downloads]# sudo rpm -Uvh chefdk-0.7.0-1.el7.x86_64.rpm
warning: chefdk-0.7.0-1.el7.x86_64.rpm: Header V4 DSA/SHA1 Signature, key ID 83ef826a: NOKEY
Preparing...
Installing...
1:chefdk-0.7.0-1.el7
Thank you for installing Chef Development Kit!
```

Figure 11: Installing Chef Developer Kit

and enhancing it for deployment. Also, in a simulated environment, we have the flexibility to use the developer kit, debugging tools, performance tools like Nagios as well as tune the set-up before taking it to an actual environment.

In this context, we need to create a simulated environment on a VMware based Linux sandbox, and then create agent nodes using Linux VM to mimic a server and node set-up, prior to installing Chef Enterprise and Chef agents.

Setting up VMware: VMware is virtualisation software to create simulated operating environments of different flavours on a single machine. For example, we can create Linux VM in a Windows machine and work in the VM from within Windows itself. It also uses all resources like printers, the Internet, networks, etc, from the host machine in the VMs.

Using virtual environments, we can create VMs of various flavours and work on simulated environments without physically installing the operating environment. This requires the VMware workstation which can be downloaded at https://my.vmware.com/web/vmware/info/slug/desktop_end_user_computing/vmware_workstation_pro/12_0

Next, download RHEL 7 ISO and install it in the VMware workstation as shown in Figure 5.

Now select the ISO image which was just downloaded to start the installation process (Figure 6).

The next step is to validate the selections and agree to proceed with the installation (Figure 7).

Once the installation is done, install all peripheral software and tools required for Chef. Since Chef is written in Ruby and uses UNIX based tools like Curl, Mail, etc, we need to install all the required software like VMware tools, for which you need root privileges (Figure 8).

This installation process will take you through a series of steps when you run *vmware-install.pl* as instructed in Figure 8, which runs as shown in Figure 9.

You must now subscribe to Red Hat Linux to enable the subscription manager to get all downloads directly in the VM itself. If you don't subscribe to the Linux installation, any tool required for Chef installation has to be manually downloaded in the host machine as an RPM package, then ported to the

```
[root@magdesklinux opscodes]# chef-server-ctl reconfigure
Starting Chef Client, version 12.5.0.current.0
resolving cookbooks for run list: ["private-chef::default"]
Synchronizing Cookbooks:
- private-chef
- enterprise
- apt
- yum
- runit
- packagecloud
Compiling Cookbooks...
```

Figure 12: Chef configuration

```
[root@magdesklinux opscodes]# chef-server-ctl test
Recipe: private-chef::ctl_config
  * template[/etc/opscodes/pivotal.rb] action create (up to date)
Recipe: private-chef::default
  * file[/etc/opscodes/chef-server-running.json] action create (up to date)

Running handlers:
Running handlers complete
Chef Client finished. 17/418 resources updated in 125.9748977953 seconds
opscodes Reconfigured!
```

Figure 13: Chef test configuration

VM, and then run manually. With the subscription option, this can be automated within the VM itself.

Now download the Chef Development Kit (DK) from <https://downloads.chef.io/chef-client/redhat/#/> and copy it to the above VM. If you have the Internet configured (proxy or direct network) in Linux VM, then use the browser in the VM (Firefox is available, by default) to download Chef Tools directly into the VM.

After installing Chef Server components (Figure 10), we can install the Chef Developer Kit (DK), which helps in writing recipes and compiling cookbooks for the Chef set-up. The screen shot in Figure 11 explains the procedure to download and install the Chef DK.

Chef configuration

Chef Server and the developer kit are now ready. Next, we need to install agents in nodes and configure them. Before starting out on this step of installing and configuring agents, let's configure the server to be fully available for use and bring it up to connect with the agents. This can be done with the *Reconfigure* option as shown in Figure 12.

We have completed the installation and configuration of Chef successfully. Now, we should validate the Chef configuration to see if it looks right and is working properly. This can be done by running a test command as shown in Figure 13. 

By: Magesh Kasthuri and Dr B. Thangaraju

Magesh Kasthuri is a senior 'Distinguished Member of the Technical Staff' in Wipro. He is a senior technical consultant in the Java Technology Practices Group and can be reached at magesh.kasthuri@wipro.com

Dr B. Thangaraju is an open source software (OSS) evangelist. He can be reached at balat.raju@wipro.com



The Five Pillars of **hadoop**

From a small side-project for a search engine, Hadoop has turned into a major force today. It is now synonymous with the term 'Big Data'. The Hadoop ecosystem has many components, which strengthen it and give it its punch.

The buzzwords these days are Big Data and Hadoop, which is there to tame it. We know that Hadoop has three components, namely HDFS, MapReduce and Yarn. HDFS stands for the Hadoop Distributed File System, which is used to store data across machines in the form of blocks, across the cluster. MapReduce is a programming model that can be used to write our business logic and fetch the required data, while Yarn acts as an interface between HDFS and other applications like Spark, HBase, etc. What we don't know is that Hadoop uses many other applications which help in its optimal performance and utilisation. In this article, I will outline the five pillars of Hadoop that make it powerful enough to operate on Big Data.

Pig

This is a platform for analysing large data sets, which consists of a high-level language that expresses data analysis programs, coupled with infrastructure for evaluating these programs. Pig is a high level language that mainly works on semi-structured data like log files. It supports the language called Pig Latin. A query planner compiles queries written in Pig Latin into maps and reduces, which are then executed on a Hadoop cluster. Using Pig, you can create your own function to do special processing. In simple MapReduce, it is really tough to write the joins between tables. This is easy in Pig because it is best at joining data sets, sorting data sets, filtering data, grouping by methods and more specifically, you can write user-defined functions (UDFs). The MapReduce programming model can be thought of as composed of three distinct phases, i.e., processing of input records, forming groups of related records and processing groups into outputs. In MapReduce, the first two of these steps is handled by the mapper, and the third step is handled by the reducer. Pig Latin exposes explicit primitives that perform actions from each phase. These primitives can be composed and reordered. Pig works in two modes—the local mode and the Hadoop mode. Local mode uses a single JVM and works on the local file system, whereas the Hadoop mode

or MapReduce mode renders Pig Latin into MapReduce jobs and executes them on the cluster.

Hive

Hive is the data warehouse of Hadoop. Those who don't have a Java background and know SQL queries, find it difficult to write MapReduce jobs in Java. To address this problem, Hive was developed. Using Hive, queries are written which get compiled to MapReduce jobs at the backend. This quickens the process as writing queries is faster than writing code. Moreover, Hive supports DDLs like *create table*, *create view*, *create index* and DMLs like *select*, *where clause*, *group by*, *order by* and *joins*. The point to remember is that Hive is not an RDBMS; it should be used for batch processing and not OLTP. Hive has its default metastore, which contains the location of table files, table definitions, storage formats, row formats, etc. Hive query is called HQL (Hive Query Language). Derby is the default database for Hive.

Sqoop

If you have data in some other RDBMS database, like Oracle or MySQL, and now want to move to using Hadoop, you have to move your data to HDFS; this is when Sqoop comes handy. Sqoop is an open source tool used for data interaction between traditional RDBMS and the Hadoop environment. Using Sqoop, data can be moved into HDFS, Hive and HBase from MySQL, PostgreSQL, Oracle, SQL Server or DB2, and vice versa. It is used widely in the industry, as it is the first Apache product to be used from the minute you decide to move from a relational DB to the Hadoop ecosystem.

Sqoop works in three steps. In the first step, it sends the request to the relational DB to return the metadata information about the table (metadata here is the data about the table in the relational DB). In the second step, Sqoop generates the Java classes from the received information, for which Java must be installed in the system. In the final step, a jar is made out of the compiled files. Sqoop needs to have

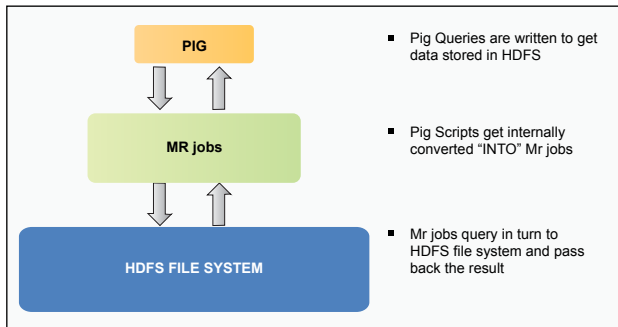


Figure 1: Pig

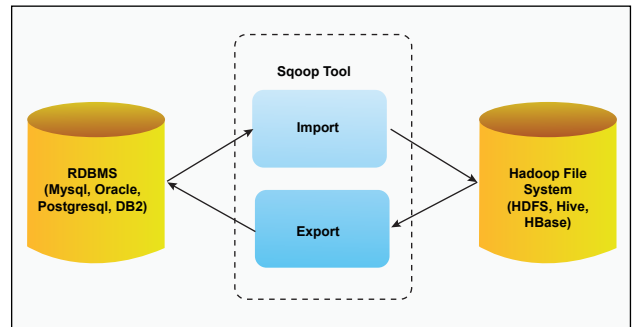


Figure 3: Sqoop

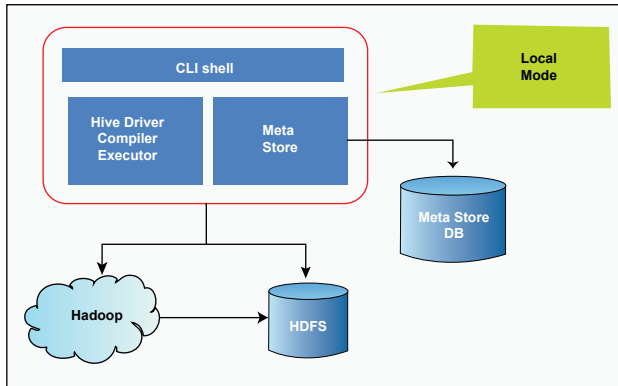


Figure 2: Hive

a primary key to work best, but do not worry if your table structure inherently doesn't have it; it will be created for you but without affecting your table's metadata structure.

HBase

HBase is a distributed column oriented database built on top of HDFS. It is the Hadoop application to use when you need real-time read/write random access to a very large data set. HBase provides APIs that enable development in practically any programming language, and is well suited for sparse data sets. It is written in Java and doesn't enforce relationships within the data. The key point in HBase is that it doesn't care about data types, storing an integer in one row and a string in another, in the same column. It stores a key value pair and stores versioned data. The HBase shell is written with JRuby (the Ruby implementation of JRE) wrapping the Java client API (i.e., one can access Java libraries). HBase runs in three different modes: standalone (running in a single JVM on a single machine), pseudo-distributed (running multiple JVMs on a single machine), and full-distributed (running multiple JVMs on multiple machines).

Zookeeper

This is a distributed coordination service for distributed applications. Problems like centralised configuration, synchronisation, partial failures, deadlocks, race conditions and networking are all solved with Zookeeper. It actually handles the nitty-gritty of distributed application development on the Hadoop ecosystem so that the developer can focus on

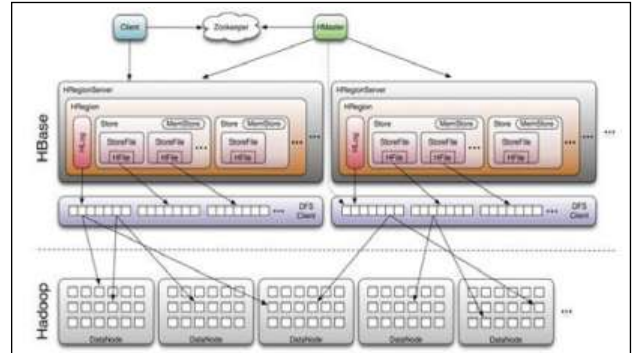


Figure 4: HBase

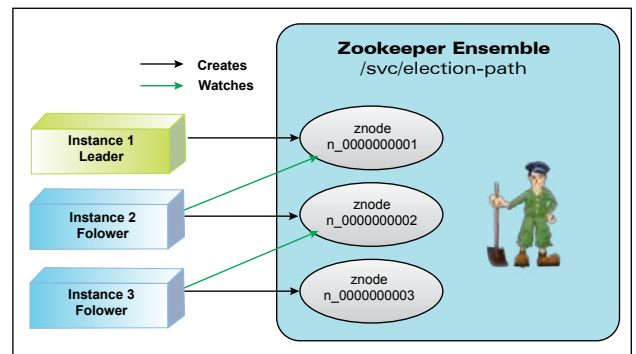


Figure 5: Zookeeper

functionality. Zookeeper always has an odd number of nodes in the cluster as the selection of the master is via voting. Zookeeper has a leader, follower and observer. In the leader, the writes are quorum based and are committed by followers. The followers forward the writes to the leader. Only one leader can write and commit to a file, and all the requests come to the leader via the followers. In case the leader goes down, voting is done among followers to select the leader. Observers just observe the results of votes but never take part in the voting process. **END** 

By: Ashish Sinha

The author is a software engineer based in Bengaluru. A software enthusiast by heart, he is passionate about using open source technology and sharing it with the world. He can be reached at ashi.sinha.87@gmail.com

WebRTC: The Way Forward for Video Over the Internet

Who doesn't love to make a free call? Web Real-Time Communication or WebRTC enables browser-to-browser communication for video calling, P2P file sharing and voice chat. This exciting, powerful and cutting-edge technology is set to revolutionise the way we communicate.



The goal of WebRTC is to provide real-time browser-to-browser voice and video communication without the need to download any plugins or software. So let's explore WebRTC architecture, the requirement for it in enterprises, and the leading market players who offer WebRTC as a service.

WebRTC is an open source project, released in May 2011 by Google, which allows browser based real-time communication. It supports browser-to-browser applications for video chat, voice calling and P2P file sharing without the need of any external or internal plugin. The WebRTC project is mainly supported by Google, Mozilla and Opera.

WebRTC components are accessed with JavaScript APIs—the Network Stream API, the PeerConnection API that allows two or more users to communicate browser-to-browser, and the DataChannel API that enables communication of other types of data for real-time gaming, text chat, file transfer, etc.

Connecting to remote peers uses NAT (network address translation) traversal technologies such as ICE (Interactive Connectivity Establishment), STUN (session traversal utilities for NAT), and TURN (traversal using relay NAT). STUN servers reside on the public Internet and have one simple task of checking the IP port address of an incoming request (from an application running behind a NAT) and sending that address back as a response. This process enables a WebRTC peer to get a publicly accessible address for itself, and then pass that on to another peer via a signalling mechanism, in order to set up a direct link. As many as 86 per cent of WebRTC calls successfully make a connection using STUN, though this can be less for calls between peers behind firewalls and complex NAT configurations.

RTCPeerConnection (PeerConnection API) of WebRTC tries to set up direct communication between peers over the User Datagram Protocol (UDP). If that fails, RTCPeerConnection resorts to Transmission Control Protocol (TCP). If that fails, TURN servers can be used

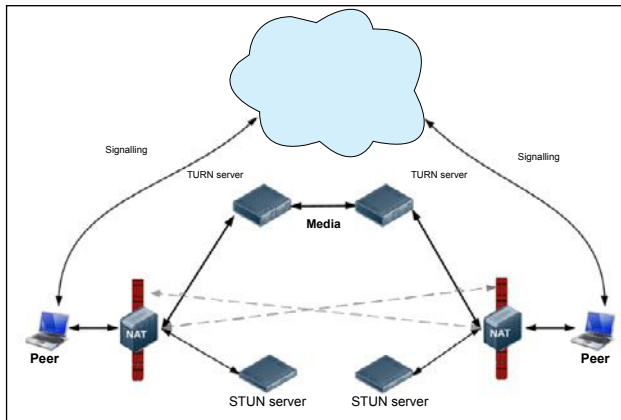


Figure 1: STUN, TURN and signalling in WebRTC

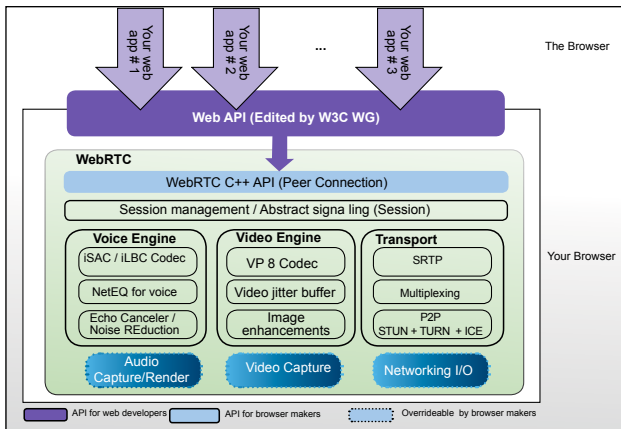


Figure 2: WebRTC architecture

as a fallback, relaying data between end points. TURN is used to relay audio, video and data streaming between peers and is not used for signalling data. TURN servers have public addresses, so they can be contacted by peers even if the peers are behind firewalls or proxies. TURN servers have a conceptually simple task — to relay a stream. For a WebRTC call, once the capabilities of the browser or application are exchanged, a point-to-point connection is established. This includes NAT traversal.

Features of WebRTC

The objective of the WebRTC project is to enable rich, high quality, RTC applications to be developed for the browser, mobile platforms, and IoT devices, and allow them all to communicate via a common set of protocols.

- WebRTC allows Web platforms to include audio conferencing, video conferencing and data sharing as part of the websites or applications
- It allows a Web browser as an end point for a call
- It allows websites access to video camera feeds to manipulate the feed or to allow users to record video
- It provides platform and device independence so that developers can write WebRTC HTML 5 code that will run across different OS browsers

- Advanced audio and video quality is provided by using the Opus audio codec and the VP8 video codec

WebRTC architecture

There are many components to the WebRTC architecture that readers need to familiarise themselves with. The most important are listed below.

Web API: The Web API is used to create Web applications, which may have video and audio chat capabilities. The Web API is used by third party developers to develop Web based video chat-like applications.

WebRTC Native C++ API: The WebRTC Native C++ API layer enables browser developers to easily implement the Web API proposal. This is used by browser developers to create WebRTC compatible browsers, and they can override the audio capture, video capture and the network I/O render.

Voice engine: This sets the framework for the audio media connection, which includes the iSAC (Internet Speech Audio Codec), the iLBC (Internet Low Bitrate Codec), the Opus codec, the NetEqualizer for Voice, the acoustic echo canceller and noise reduction technologies. The iSAC / iLBC / Opus audio codec for VoIP and audio streaming over wideband and narrowband supports constant and variable bitrate encoding from 6 kbits/s to 510 kbits/s, as well as frame sizes from 2.5 ms to 60 ms.

The NetEQ for Voice is a dynamic jitter buffer and error concealment algorithm used for concealing the negative effects of network jitter and packet loss, which keeps latency as low as possible while maintaining the highest voice quality.

The acoustic echo canceller is a software based signal processing component that removes the acoustic echo resulting from the voice being played out coming into the active microphone. The noise reduction component is a software based signal processing component that removes certain types of background noise usually associated with VoIP, such as a hiss, fan noises and other background disturbances.

Video engine: This is the framework for the video media connection for video, from the camera to the network, and from the network to the screen and back. It includes the VP8 (video compression format owned by Google) codec, dynamic jitter buffer and image enhancement technology. The VP8 codec by the WebM Project (an open media file format designed for the Web) is well suited for RTC as it is designed for low latency. The dynamic jitter buffer for video helps conceal the effects of jitter and packet loss on overall video quality, and image enhancements remove video noise from the image capture by the webcam.

Transport session: Transport session components are built by re-using components from libjingle, without using or requiring the Extensible Messaging and Presence Protocol (XMPP) or Jingle Protocol. The Real Time Protocols (RTP) network stack and the STUN and ICE components allow calls to establish connections across various types of networks.

The need for WebRTC in the enterprise

Open source technology from Google, which is now a standard in development with the IETF (Internet Engineering Task Force), could lead to inexpensive and interoperable video solutions in the business world, where large enterprises have been forced to invest hundreds of thousands of dollars in collaboration architecture, and smaller companies couldn't even consider the technology due to price and complexity.

Sending data between two users in today's browser world is a tough process with JavaScript, and most developers rely on a server as the middle agent. With WebRTC, the Data Channel API allows the passing of arbitrary data across the connection. This allows for large scale file sharing, fast action multiplayer games, and even remote control applications. The Data Channel allows users to transfer large amounts of data very quickly, all in the Web browser in true peer-to-peer mode.

WebRTC adds advanced levels of interaction to a website, allowing users to communicate in-context and in real-time, with either site operators or with each other. Such capabilities overcome current limitations that require users to call a number, download a plugin, or leave the website.

The uses of WebRTC

WebRTC use-cases can range from basic video/voice chat, to multiplayer gaming, file sharing and so on. Listed below are some of the possibilities.

Video recruitment and interviewing

Video interviews and recruiting have become more popular as enterprises and small businesses try to make the hiring process more efficient. Companies could conduct video interviews just by sharing a URL with the candidate and thereby decrease the cost of hiring. They could screen candidates faster, and the video session could be recorded and shared for further review.

Live video technical app troubleshooting support

Video customer support for Web or mobile applications gives customers a better user experience. The customer support executive could interact and troubleshoot issues with users on their device. By using WebRTC's screen sharing, customer service agents could save time while troubleshooting with customers. Users could grant real-time device control to the support team to troubleshoot and eliminate confusing communication barriers.

Field service video calling

In the field, service technicians could use the camera on smartphones or tablets to contact experts located at a remote location to get help on troubleshooting devices and equipment, eliminating the need to send an expert to a remote location. This would reduce service downtime, decrease

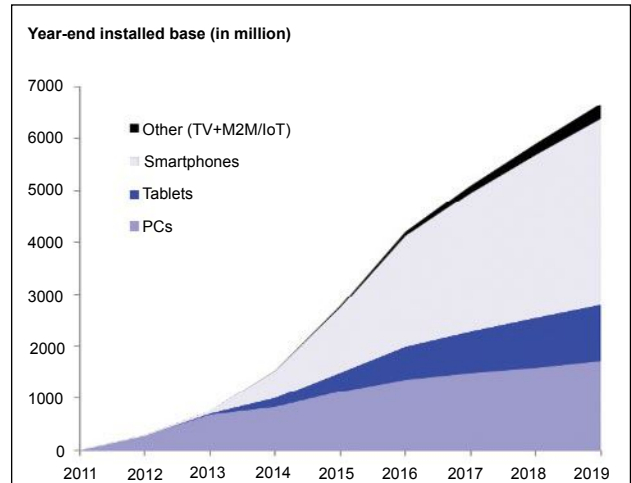


Figure 3: WebRTC market forecast

operational costs and improve customer satisfaction.

Online shopping assistants and concierge services

Online retail shopping assistants and concierge services could use video calls to increase brand loyalty, sales and customer satisfaction. When customers seek advice on an outfit or an accessory for their outfit or when an agent is consulted on the use of makeup to complement the client's features, there is increased customer satisfaction. The common occurrence of abandoning the shopping cart can be reduced by connecting with customers in real-time and thereby increasing sales as well as brand loyalty.

Equipment or product troubleshooting

Customers could use a video to show a customer service agent issues with a specific item and configuration before sending or returning the product. This could save time while troubleshooting with customers, and also reduce processing and shipping costs for returning the product.

Market trends

Since the initial release of the WebRTC API by Google in May 2011, the project has grown both in terms of support and interest. According to Dean Bubley's 'Disruptive Analysis', nearly three billion devices are WebRTC-capable today, with an estimate of about 6.5 billion devices supporting WebRTC by 2019. Today, desktop browsers lead in WebRTC usage, with Google Chrome, Mozilla Firefox and Opera accounting for more than 60 per cent of total browser usage. In the years to come, usage will increase quickly with the availability of browser support on smartphones.

By 2018, there will be an interesting traction for WebRTC within the Internet of Things (IoT) space. By definition, IoT connects things to the Web to interact with them in real-time. So, it makes sense that IoT, combined with WebRTC, can create a promising future with real-time communications.

IGATE's Live Remote Assistance app enabled by wearable smart glasses

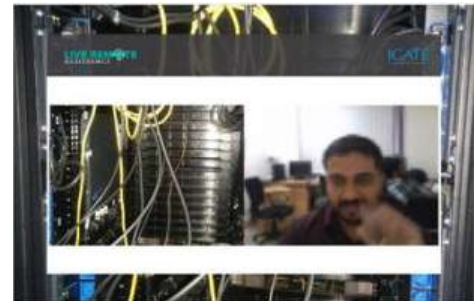
During the field visits of a service engineer, referring to a product manual or using mobile assistance to service particular equipment might not work due to the inexperience of the engineer. Experts in a remote location, who do not have a direct view of the equipment or device being serviced, will not be helpful too. To overcome these difficulties, IGATE has developed a 'Live Remote Assistance' wearable glass application, that provides the assistance of a remotely located expert to field technicians. This application can be used in manufacturing, logistics, utilities and other industries.

IGATE has used Epson Moverio BT-200 wearable smart glasses, and the application makes use of its camera through which a live feed is sent over the Internet to the remote expert, who then guides the user by voice and video. The expert can provide better assistance to the field technicians with a live view provided by the wearable glass camera.

Video and voice collaboration with experts in remote locations using Live Remote Assistance would result in faster repairs and save the expense of flying in an expert to the site for help. These collaborations can be recorded and stored as reference material for future jobs, as well as video evidence for use in future in-depth investigations.



Live Remote Assistance for network troubleshooting



Live Remote Assistance wearable glass view

Leading market players offering WebRTC as a service

As WebRTC has gained wider popularity, the opportunity to offer it as a service has risen. Featured here are some of the more well-known options.

PubNub

PubNub offers a data stream network that can be used as a scalable signalling server for WebRTC applications. PubNub features Presence and storage/playback, which can be used to enhance applications.

The WebRTC protocol does not provide storage capabilities, and as a result, there are no records of what messages have been sent. Specifically with text chat, users expect a history of previous chat conversations. PubNub's storage/playback feature allows users to see a history of past conversations over a desired period of time.

Bistri

Bistri was the very first service to provide a production platform for the implementation of WebRTC applications. Bistri harnesses WebRTC to provide users with a personalised link that, when clicked, instantly initiates a video call, like an online phone number. There is no software, plugins or accounts required; the call is carried out completely through the browser. A single snippet of JavaScript is all that is needed to attach a button to a website or blog, offering visitors a simple click-to-video-call experience.

TokBox

TokBox is a PaaS (Platform as a Service) company that provides hosted infrastructure, APIs and the tools required to deliver enterprise-grade WebRTC capabilities. It does so primarily through its proprietary OpenTok video platform for commercial application.

The OpenTok platform provides APIs, a global cloud infrastructure and pre-configured solutions to enterprises, entrepreneurs and developers. Originally launched as an Adobe Flash based platform in 2010, OpenTok became the first real-time communications platform to incorporate support for WebRTC in 2012. In mid-2013, TokBox launched Mantis, a cloud based infrastructure enabling multi-party conversations on the OpenTok platform.

Developers use the OpenTok platform to deploy WebRTC applications with 10x fewer lines of code than required through WebRTC off-the-shelf. OpenTok also supports multi-party calling, calls between browsers and iOS or Android devices, call archiving and playback, secure enterprise firewalls traversal and more.

Challenges in adopting WebRTC

Implementation of current WebRTC specifications has not yet been completed by different browser providers. Google Chrome, Firefox and Opera support WebRTC without any plugins. Chrome v29 or greater supports WebRTC on Android. Internet Explorer and Safari don't support WebRTC natively. Third party plugins can enable WebRTC features on these browsers to an extent.

Another challenge with WebRTC is the unfortunate codec battle. Google, Mozilla and the W3C support VP8 mostly due to their belief that the video codec should be royalty-free, allowing developers to use WebRTC without having to pay licence fees. Apple, Microsoft and Cisco favour H.264, since it is more prevalent and currently many devices have H.264-optimised hardware for accelerated encoding and decoding.

WebRTC is the future technology standard for businesses, as it enables users to communicate more easily and customers to get answers quickly with click-to-call features in the website or in applications. Businesses that embrace WebRTC could gain commercial benefits and a competitive advantage due to better customer engagement, boosted sales, improved communication and reduced call costs.

WebRTC technology is widely supported and is being actively developed. End users and enterprises are adopting WebRTC because of its advantages and simplicity of use. One of the challenges of WebRTC in adoption was browser support. Now, even the non-supported browsers are releasing plugins and updates to support it. In the near future, when the market will feature more high quality, user-ready solutions, the popularity of WebRTC will only increase.

Glossary

P2P	Peer-to-peer
NAT	Network address translation
ICE	Interactive Connectivity Establishment
STUN	Session Traversal Utilities for NAT
TURN	Traversal Using Relay NAT
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
VP8	VP8 is a video compression format owned by Google



References

- [1] WebRTC (Web Real-Time Communication), Wikipedia, 2015 <http://en.wikipedia.org/wiki/WebRTC> (Accessed – June 2015)
- [2] WebRTC, Mozilla Developer Network, 2015 <https://developer.mozilla.org/en-US/docs/Web/Guide/API/WebRTC> (Accessed – June 2015)
- [3] 'WebRTC in the real world: STUN, TURN and signalling,' Sam Dutton, HTML5Rocks, 2013 <http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/> (Accessed – June 2015)
- [4] <http://www.webrtc.org/> (Accessed – June 2015)
- [5] WebRTC (Web Real-Time Communication), Wikipedia, 2015 <http://en.wikipedia.org/wiki/WebRTC> (Accessed – June 2015)
- [6] <http://www.pubnub.com/blog/what-is-webrtc/> (Accessed – June 2015)
- [7] 2015 Q1 Update: WebRTC Market Status & Forecasts Report, Disruptive Analysis, 2015 http://disruptivewireless.blogspot.in/p/blog-page_30.html (Accessed – June 2015)
- [8] 'Are We at the Tipping Point of WebRTC Adoption?' AT&T website, 2015 <http://developerboards.att.lithium.com/t5/AT-T-Developer-Program-Blogs/Are-We-at-the-Tipping-Point-of-WebRTC-Adoption/ba-p/40202> (Accessed – June 2015)
- [9] TokBox, Wikipedia, 2015 <https://en.wikipedia.org/wiki/TokBox> (Accessed – June 2015)
- [10] 'SightCall Use cases,' <http://www.sightcall.com/use-cases/> (Accessed – June 2015)

By: Vijay Sanjos Alexander and Venkatesh Babu

Vijay Sanjos Alexander is a technical specialist with more than seven years of experience in the IT industry. At IGATE, he is currently working in the 'Technology CoE in Research & Innovation' group, on applications for emerging technologies and trends.

Venkatesh Babu is a senior principal architect working in the 'Technology CoE in Research & Innovation' group at IGATE. He has more than 20 years of experience in the IT industry and is currently involved in delivering applications of emerging trends and technologies.

OSFY Magazine Attractions During 2015-16

MONTH	THEME	BUYERS' GUIDE
March 2015	Open Source Firewall and Network security	SSD for Servers
April 2015	Web Development	Network Switches
May 2015	Virtualisation (containers)	Wireless Routers for SME
June 2015	Open source Databases	PaaS Solution
July 2015	Network Monitoring	MFD Printers for SMEs
August 2015	Mobile App Development	Hosting Solutions
September 2015	Backup and Data Storage	External HDD
October 2015	Programming Language	IaaS Solution
November 2015	Cloud Special	Firewall and UTMs
December 2015	Open Source on Windows	Online Backup solutions
January 2016	Android Special	Wifi Hotspot Devices
February 2016	Top 10 of Everything	External Storage

Building Apps with Meteor

Meteor is a powerful platform for Web app development and an interface at which the mobile world integrates with the Web world. The 'universal language everywhere' feature of Meteor makes it extremely simple for developers to write apps. So it's an option that's certainly worth considering for Web and mobile app development.



Meteor is an open source platform which provides a complete, full-stack framework for building Web and mobile applications entirely in JavaScript. It provides a very easy and delightful app-building experience with lots of cool features such as live updates, a unified package system, Websocket Microservices, etc. Meteor has a reactive programming model that extends all the way from the database to the user's screen. So let's explore how to build an app with Meteor.

Installing Meteor

Installing Meteor is pretty straightforward. There is just one command to be executed, after which you are all set. To install Meteor on OS X or Linux, run the following command from the terminal.

```
$ sudo apt-get install curl  
$ curl https://install.meteor.com/ | sh
```

To install Meteor on Windows, just download the installer from <https://install.meteor.com/windows>, follow the wizard and it's all done.

This will install Meteor, MongoDB and almost everything you need to get started with.

Creating a new app

To create a new app, run the following command from the terminal:

```
$ meteor create meteorApp
```

This creates a new directory named *meteorApp* and places all the files in it.

Inside the *meteorApp* directory, the following three files are created by Meteor:

1. *meteorApp.js* contains all the JavaScript code which will be sent to both client and server
2. *meteorApp.html* is an HTML file to define views

3. *meteorApp.css* is a CSS file to define styles

To run the app, first navigate to the app directory and run the Meteor command, as follows:

```
$ cd meteorApp
$ meteor
```

Now open the browser and enter the URL *http://localhost:3000*, where you will see output as shown in Figure 1.

Structure

By default, any JavaScript file found by Meteor is automatically loaded and run by it, both on the client as well as the server side. This can be a serious issue in terms of security and hence organising the code into appropriate directories is very important. Meteor provides a special way to have control of the JavaScript code to be loaded on the client and the server by using special directories which isolate the code on the server and the client side. The following is a list of special directories in Meteor:

/client

All the files in this directory are sent to the client and are published to the browser. HTML, CSS and related JavaScript can reside here.

/server

Files within this folder are run on the server only.

/public

Files in this directory are served to the client. Assets like images and videos can reside in this directory which serves as the root for assets, i.e., when including files from this directory the directory name */public* is not a part of the URL.

/private

Files within this folder can be used by the server using the assets API.

If any HTML and CSS file is encountered outside these directories, it's sent to the client side while the JavaScript gets loaded both on the client and the server side.

Templates

Templates are nothing but Meteor's way of specifying HTML. Clients render HTML through templates. Templates also serve as a connection between the interface and the JavaScript code. All templates in Meteor are defined inside the template tag.

Here's an example:

```
<template name="foo">
.
..
...
</template>
```

With the *name* attribute, we define a name for the template by which it is referred.

For every defined template, Meteor creates a template



Figure 1: Default Meteor output

object. Everything inside a *<template>* tag is compiled into Meteor templates, which can be included inside HTML with *{{> templateName}}* or referenced in your JavaScript with *Template.templateName*.

Let us proceed to our first template by issuing the following code:

```
<template name="foo">
<h1> Welcome to Meteor</h1>
</template>
```

Now we need to include this template inside the interface. To make the 'foo' template appear on the page, place the following line inside the *.html* file:

```
{{> foo}}
```

Given below is an example of how to create and embed the template inside the HTML body.

```
<head>
  <title> Meteor </title>
</head>
<body>
  <h1> Hello World </h1>
  {{> foo}}
</body>
<template name="foo">
<h1> Welcome to Meteor</h1>
</template>
```

To view the above code in the browser, first run the Meteor server.

In the browser, enter the URL *localhost:3000* and you will see the results, as shown in Figure 2.

Templates and JavaScript

Templates can become more dynamic with the use of helpers and they can be referenced in JavaScript using *Template*.

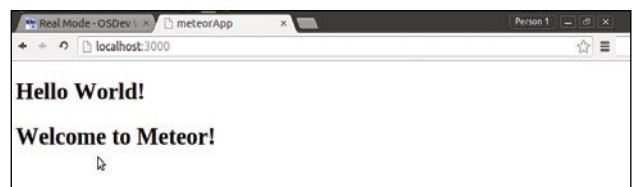


Figure 2: Defining template

templateName, as follows:

```
<body>
<div class="container">
  <ul>
    {{#each lines}}
      {{> line}}
    {{/each}}
  </ul>
</div>
</body>
<template name="helperDemo">
  <li>{{line}}</li>
</template>
```

helperdemo.js: The following helper passes data to the template. The helper named 'lines' returns an array to the template, where we loop over the array and display each line in an unordered list using html tags.

```
Template.body.helpers({
  lines: [
    { line: "This is line 1" },
    { line: "This is line 2" },
    { line: "This is line 3" }
  ]
});
```

Collections

Collections are nothing but databases. By default, Meteor comes with MongoDB, which is automatically installed and doesn't require any configuration. This database can be accessed on the server as well as the client side.

To create a collection, just add the following line in a js file:

```
MyDb = new Mongo.Collection("testDb");
```

This database can be viewed both from the server and the client. On the server side, the database operations can be performed on the Meteor Mongo console and on the client side, these can be done on the browser console. In order to insert data into the database, open a new terminal tab, go to your *app* directory and type the following command:

```
meteor mongo
```

This opens a console in your app's local development database. At the prompt, type the following command:

```
db.testDb.insert({ text: "Hello world!", createdAt: new Date() });
```

The same can be done with the browser console or through the JavaScript file. Now this database is automatically

made available to both client and server. To view the inserted elements, enter the following command:

```
db.testDb.find();
```

Events

In Meteor, events are associated with templates. Events are attached to templates in the form *Template.templateName.events*, where *events* is a keyword. The following code attaches a click event to a template.

events.html

```
<head>
  <title>Meteor events</title>
</head>
<body>
  <h1>Welcome to Meteor!</h1>
  {{> meteorEvents}}
</body>
<template name="meteorEvents">
  <button>Click Me</button>
</template>
```

events.js

```
Template.meteorEvents.events({
  'click button': function () {
    console.log("You have clicked a button")
  }
});
```

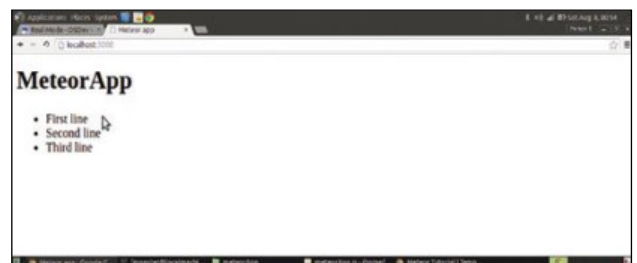


Figure 3: Defining *Views* with templates

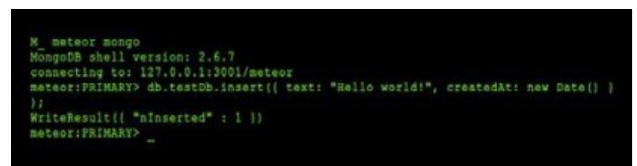


Figure 4: Inserting values into MongoDB

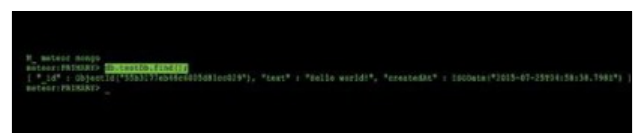


Figure 5: Fetching values from MongoDB

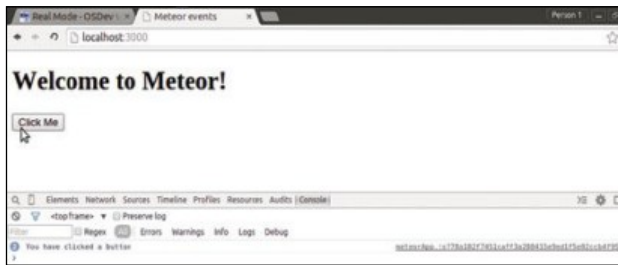


Figure 6: Meteor events

Meteor also supports all types of HTML DOM events.



Note: `preventDefault()`: prevents the action the browser would normally take in response to this event, such as following a link or submitting a form. Further handlers are still called, but cannot reverse the effect.

Methods

Meteor methods are like functions in JavaScript with a different syntax. They can be called by the client as well as by the server.

Let us now look at how to create methods in Meteor.

```
Meteor.methods({
```

```
'foo': function(){
  console.log("Hello world, I am Method");
}
});
```

Every method has a method name and is associated with a function, which is executed when the method is called.

The `Meteor.call()` method is used to trigger the execution of the method. We pass a parameter in the `Meteor.call()` method which is the name of the method to be called. Shown below is the syntax to call the method `foo`.

```
Meteor.call('foo');
```

Meteor methods can accept parameters and also return a value. **END** 

References

[1] www.meteor.com

By: Randeep Singh Monga

The author is a final year student in information systems studying at BITS Pilani, Pilani Campus. He is also a software developer, currently interning at Altair Engineering in Bengaluru. He can be reached at monga.randeep@gmail.com

India's #1 Website For **Electronics** Engineers Working Across The Globe

Established: **1998** | Registered Users: **300,000+**

Monthly Statistics

Page Impressions: 750,000+

Unique Techies: 150,000

electronicsforu.com

If it's electronics, it's here



For any queries, please contact our team at efyenq@efy.in OR +91-11-26810601

An **EFYGROUP**
PROPERTY



OSI 2015 Tracks

Mobile App. Dev Day

Cloud Day

IT Infrastructure Day

Kernel Day

Database Day

FOSS for Everyone

IT Implementation Success Stories

Web App. Dev Day

OpenStack Mini Conference





Is the **Closed**
Code
Taming the
'Creative
Lion'
within you?

Come & Discuss **OPEN SOURCE**



12th Edition

OPEN
SOURCE INDIA

19 - 20
November
2015

NIMHANS Convention Center

BENGALURU

Asia's largest
conference on
Open Source

2 Days. 9 Tracks. 8+ Workshops. 40+ Speakers

Android Studio: A Platform for Android Development

Android Studio is an integrated development environment (IDE) for the Android platform. It simplifies app development. Though offered by Google, seasoned Java developers will immediately recognise that the toolkit is a version of IntelliJ IDEA.

According to IDC, globally, Android's share of the smartphone market is about 45 per cent. The best part is that Android is open source and learning it is not at all difficult. Students and professionals want to, at least, know its basics. There are many platforms, like Android Studio, where even beginners can get into Android development. Android Studio is a cross-platform integrated development environment (IDE) for developing on the Android platform. It is written in Java and is available for Linux, Windows as well as for Mac OS X.

Eclipse, which also provided Android development tools, has been replaced by Android Studio as Google's primary IDE for native Android application development. The main reason for this move is because Eclipse was not stable.

Android Studio offers a better *Gradle* build environment, smarter short cuts, an improved user interface (UI) designer, a better memory monitor, an improved string translation editor and better speed. The build system in Android Studio replaces the Ant system used with Eclipse ADT. It can run from the menu as well as from the command line. It allows you to track memory allocation as it monitors memory use. It has built-in support for the Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine. It also comes with inline debugging, and performance analysis tools.

Android Studio has Android Virtual Device (AVD) which comes with emulators for Nexus 6 and Nexus 9 devices. It also offers build variants and the capability to generate multiple *apk* files. Whenever one compiles a program, the configured *lint* and IDE inspections run automatically.

Configuration

Installation

Before you set up Android Studio in Linux, you need to install JDK 6 or higher. In fact, JDK 7 is required for developing Android 5.0 or above. The other requirements are a minimum of 2GB RAM (though 4GB is recommended), 400MB hard disk space and at least 1GB for the Android SDK, emulator system images, caches, GNU C Library (glibc) 2.15 or later, etc. After installing Android Studio and setting it up, go to the SDK manager to update the required tools, platforms, etc, required for app-building. These packages provide the basic SDK tools for app development, without an IDE. If you prefer to use a different IDE, the standalone Android SDK tools can be downloaded. One can set up an update channel to *Stable* by going to: *File > Settings > Appearance & Behaviour System Settings > Updates* as shown in Figure 1.



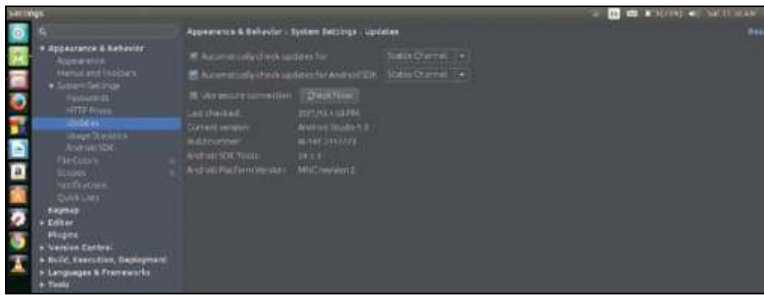


Figure 1: Getting updates

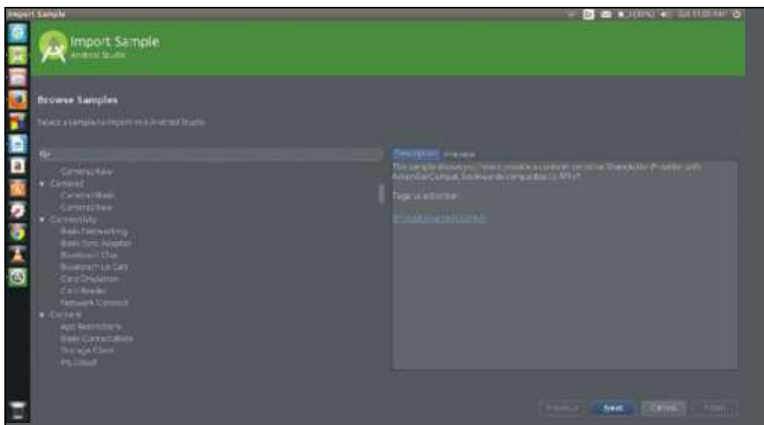


Figure 2: Importing samples from GitHub



Figure 3: Modules for different requirements

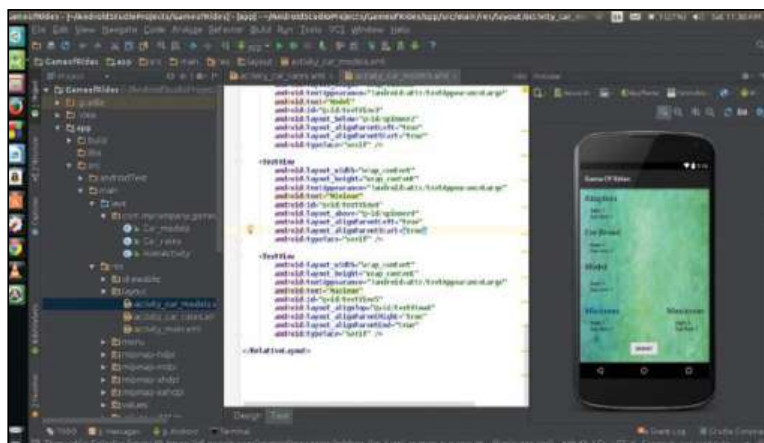


Figure 4: Preview of the app

Proxy settings

One needs to set the proxy settings for the IDE and the SDK manager in order to support Android Studio to run behind the firewall.

For Android Studio:

- From the main menu, choose: *File > Settings > Appearance & Behaviour—System Settings—HTTP Proxy.*
- In Android Studio, open the *IDE Settings* dialogue box. Choose: *File > Settings > IDE Setting -- HTTP Proxy.*
- When the HTTP proxy page appears, select auto-detection to automatically configure the proxy settings or manually enter each of the settings.
- Click *Apply* to enable the proxy settings.

For the SDK manager:

The SDK manager's proxy settings enable proxy Internet access for Android package and library updates from the *SDK manager* packages. Start the SDK manager and open the *SDK Manager* page.

- Choose *Tools > Options* from the system menu bar.
- The *Android SDK Manager* page appears. Enter the settings and click *Apply*.

Features

- Code samples are available on GitHub. To import them, go to *File > Import Samples* as shown in Figure 2. This would be very useful for newbies.
- Android Studio also supports Android Wear and T.V. While creating a project, you can select features according to the requirements. For that, go to *File > New > New Module* (Figure 3).
- Template based wizards are used to create and manage activities within the project. This tool helps the developer choose which API level to target by displaying information about the Android version name and number, the distribution and the APIs present.
- It has a rich layout editor that helps in adding UI components by just dragging and dropping. The layout can be previewed for various screen orientations providing cross-platform consistency. It also shows the preview of the app (Figure 4).
- It supports fingerprint authentication for the app.
- The android manager page appears. Enter the settings and click apply

Using hardware devices

After you have successfully created a project, it can be run either on a virtual device (emulator) or on your own device. Though seeing your app run on your own device can be fun, the advantage of running it on the emulator is that it can be verified on different versions of the Android platform. You also have the option to check how it functions in different orientations and screen sizes. To run the app on your device:

- First connect the device to the development machine through a USB cable.
- Go to *Settings > Developer Options* on your device. If you use Android versions higher than 4.2, the developer options are hidden. So, to activate them, go to *Settings > About phone* and tap *Build number* seven times. Then go back to the previous menu where you will now find the *Developer Options*.
- Enable the *Debugging over USB* option.
- Now, go to your project and select one of your project's files and click *Run* from the toolbar.
- In the *Choose device* window that appears, select the *Choose a running device* radio button, select your device, and click *OK*.

Building and running a project

Before you can run your application on the Android emulator, verify the default AVD or create one. To run (or debug) your application, select *Run --> Run (or Run --> debug)* from the Android Studio menu bar. It will compile the project, create a default configuration, install and finally run on an emulator. If you run the application in debug mode, the *Choose a Device* option appears so you can select an attached device or emulator.

Running from the command line

To build in debug mode, open a terminal and navigate to the root of your project directory. Use Gradle to build your project in debug mode, invoke the *assembleDebug* build task using the Gradle wrapper script, as follows.

```
$ chmod +x gradlew
$ ./gradlew assembleDebug
```

To see a list of all the available build tasks for your project, type the following command:

```
$ ./gradlew tasks
```

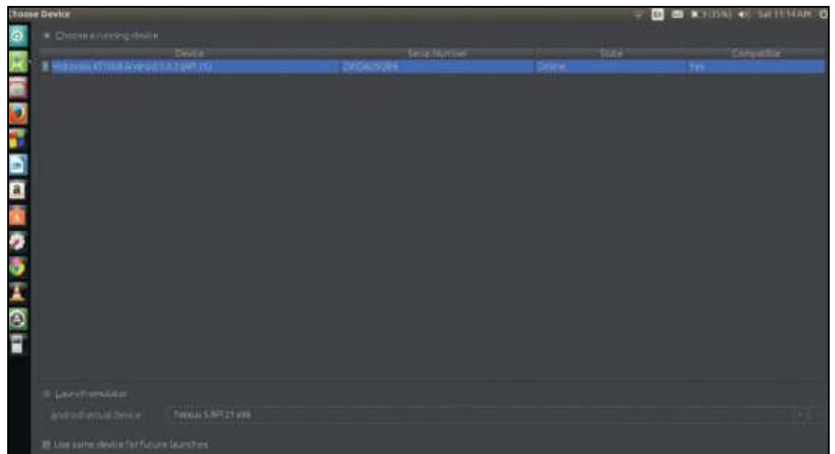


Figure 5: Selecting an external device

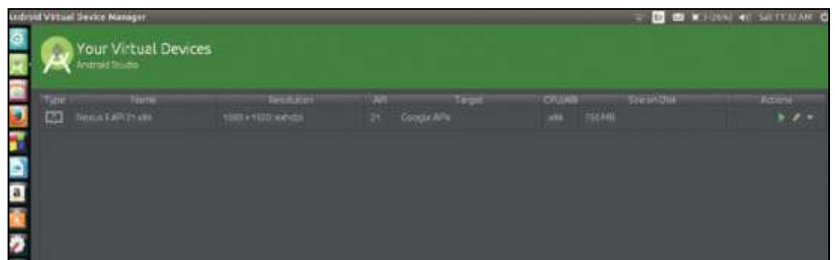


Figure 6: Selecting a virtual device (Nexus 5)

Tips

- Here are few tips for newbies. The following short cuts can save a lot of time and help you work efficiently.
- It would be very easy if when you missed adding attributes to your button or text view, a rendering message gets displayed, which, when clicked upon, would automatically add all the missing attributes.
- While creating a new project/activity, you can select the type of activity you want. It has very smart auto-code completion. Just pressing *Ctrl+space* will enable this feature.
- Method documentation can be obtained by hovering your cursor over the method definition, and clicking and pressing *F1* will bring a pop-up box with the required details.
- *Alt+Enter*: This helps in fixing coding errors like missing imports, variable assignments, etc.
- When a reference to an image is inserted in the code, a preview appears in the margin. Pressing *F1* with the preview image selected displays the resource asset details. **END** 

References

- [1] <http://developer.android.com/intl/zh-tw/tools/studio/index.html>

By: Sakshi Vaid

The author is a FOSS enthusiast and loves learning about the latest technologies. She is presently working on Android and blogs at sakshivaid.wordpress.com

Using Python for the Cloud and Big Data Analytics

The classical programming languages are not effective in fetching live streaming data from the Internet. Here's where versatile programming languages come to the rescue. Read on to understand how Python can be used for the cloud and for Big Data analytics.



As per international statistical reports, every day, WhatsApp gets around 1 million new user registrations and has 700 million active users. Around 30 billion messages get sent and 34 billion messages are received daily (source: *statista.com*). Twitter statistics reveal 350 million tweets daily and more than 500 million accounts. Data is growing at a rapid rate every moment, and there are predictions that the production and generation of data in 2020 will be 44 times more than 2009 levels (source: <http://wikibon.org/blog/big-data-statistics/>).

This data is unstructured in nature, which means that it is in different and heterogeneous formats. Such vast volumes are typically known as Big Data. Deep investigation of intelligent and meaningful patterns from such data is known as Big Data analytics. A number of researchers and scientists are working

in this domain using assorted technologies and tools. There are a number of ways in which live data can be obtained for research and development. One of these is getting data from 'open data portals'. These portals provide authentic data sets for research and development in multiple domains, which can be downloaded in multiple formats including XML, CSV, JSON and many others.

Prominent portals for fetching open data

- Databhub is available at <http://databhub.io/>
- Book-Crossing - <http://www.informatik.unifreiburg.de/~chiegler/BX/>
- World Health Organization - <http://www.who.int/research/en/>
- The World Bank - <http://data.worldbank.org/>
- NASA - <http://data.nasa.gov/>

- United States Government - <http://www.data.gov/>
- Machine Learning - <http://bitly.com/bundles/bigmlcom/2>
- Scientific Data University of Muenster - <http://data.uni-muenster.de/>
- Hilary Mason research-quality - <https://bitly.com/bundles/hmason/1>

The other way to get data sets is to generate our own data sets using programming languages. The major issue here is the selection of a suitable programming language or tool that can fetch live data from social media applications or live streaming portals. The usual programming languages are not effective in fetching live streaming data from the Internet.

Python is one of the outstanding and efficient programming languages that can communicate with the live streaming servers. You can use it to store the fetched data in the database or file system for analysis and predictions.

Let's move on to look at some real-life cases in which Python has been used to fetch live streaming data.

Web scraping using Python scripts

Python scripts can be used to fetch live data from the Sensex. This technique is known as Web scraping.

Figure 1 gives a screenshot of the live stock market index from *timesofindia.com*. The frequently changing Sensex is fetched using Python and stored in a separate file so that the record of every moment can be saved. To implement this, the library of *BeautifulSoup* is integrated with Python.

The following code can be used and executed on Python. After execution, a new file named *bseindex.out* will be created and second-by-second Sensex data will be stored in it. Once we flood the live data in a file, this data can be easily analysed using data mining algorithms with SciLab, WEKA, R, TANAGRA or any other data mining tool.

```
from bs4 import BeautifulSoup
import urllib.request
from time import sleep
from datetime import datetime
def getnews():
    url = "http://timesofindia.indiatimes.com/business"
    req = urllib.request.urlopen(url)
    page = req.read()
    scraping = BeautifulSoup(page)
    price = scraping.findAll("span", attrs={"class": "red14px"})
    [0].text
    return price
with open("bseindex.out", "w") as f:
    for x in range(2,100):
        sNow = datetime.now().strftime("%I:%M:%S%p")
        f.write("{0}, {1} \n ".format(sNow, getnews()))
        sleep(1)
```

Using a similar approach, YouTube likes can be fetched and analysed using Python code, as follows:

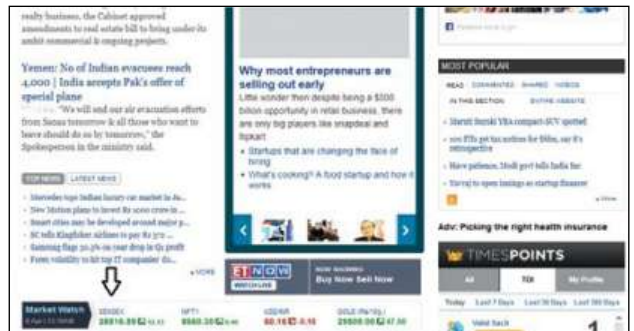


Figure 1: Screenshot of *timesofindia.com* to analyse Sensex attributes



Figure 2: A screenshot of the *Baahubali* movie trailer from YouTube

```
from bs4 import BeautifulSoup
import urllib.request
from time import sleep
from datetime import datetime
def getnews():
    url = "https://www.youtube.com/watch?v=VdafjyFK3ko"
    req = urllib.request.urlopen(url)
    page = req.read()
    scraping = BeautifulSoup(page)
    price = scraping.findAll("div", attrs={"class": "watch-view-count"})[0].text
    return price
with open("baahubali.out", "w") as f:
    for x in range(2,100):
        sNow = datetime.now().strftime("%I:%M:%S%p")
        f.write("{0}, {1} \n ".format(sNow, getnews()))
        sleep(1)
```

Fetching all the hyperlinks of a website

All the hyperlinks of a website can be fetched by using the following code:

```
from bs4 import BeautifulSoup
import requests
newurl = input("Input URL")
record = requests.get("http://" + newurl)
```

```
mydata = record.text
mysoup = BeautifulSoup(mydata)
for link in mysoup.find_all('a'):
    print(link.get('href'))
```

Using Python in cloud infrastructure

There are a number of service providers offering private or public cloud services. Listed below are a few prominent cloud and Big Data based service providers that provide options to code and deploy one's own applications:

- Amazon EC2
- IBM Bluemix
- Microsoft Azure
- Google
- Qubole

Cloud service providers make use of specialised packages and tools for coding in different languages. Python can be used on these cloud computing infrastructures.

On IBM Bluemix, the cloud services can be accessed using <https://www.bluemix.net>, and can be used once the credentials are verified. Cloud infrastructure can be used with the Cloud Foundry CLI (Command Line Interface).

After installation of Cloud Foundry (CF), one can communicate with the remote server. To check the version, the following command is used:

```
$ cf -v
```

The following instruction is used for pushing the application on the cloud server:

```
$ cf push MyApp -m 128M -b https://github.com/cloudfoundry/cf-buildpack-python.git
```

Programming Python for NoSQL databases

NoSQL databases are being used in social media applications and portals that process Big Data—in which huge, heterogeneous and unstructured data formats are handled. NoSQL databases are used for faster access of records from the big data set at the back-end. The *Aadhaar* card implementation in India is being done using NoSQL databases as huge amounts of information are involved, including text data, images, thumb impressions and iris detection. No traditional database system can handle data sets of different types (text, video, images, audio, thumb impressions, iris samples, etc) simultaneously.

Currently, a number of NoSQL databases are used for different types of portals, and these specialise in handling heterogeneous and unstructured data.

In traditional Web based implementations, RDBMS packages are deployed for database applications including Apache Derby, MySQL, Oracle, IBM DB2, Microsoft SQL Server, IBM Notes, PostgreSQL, SQLite, Sybase, etc. These are known as traditional SQL databases that comply with the ACID (atomicity, consistency, isolation and durability) properties. NewSQL is the

new-generation database engine that provides the scalability and performance of NoSQL systems for online transaction processing (OLTP) read-write workloads, while maintaining the ACID guarantees of a classical database system.

Nowadays, Web applications use unstructured data in heterogeneous formats including audio, video, text, streaming, signals, images, pixels and many others. In each file pattern, there are a number of file formats. For instance, in video, there are a number of file formats including MPEG, MP4, AVI, 3GP, WMV, OGG, FLV and others. In the same manner, image or graphics file formats include GIF, PNG, JPEG, PCX, BMP, TIFF and lots of others.

The major issue here is the compatibility of Web applications with all these file formats in different domains. Here's where the concept of NoSQL databases comes into play since they enable any type of file format to be processed and integrated in the Web applications. A NoSQL (Not Only SQL) database provides the system for storing and retrieving data, and is not modelled on the tabular relations methodology used in relational databases. The data structure in NoSQL databases is entirely different from that in traditional RDBMSs. Currently, the former are being rapidly adopted in Big Data and real-time Web applications.

There have been various approaches to classifying NoSQL databases, each with different categories and sub-categories. Because of the variety of approaches and overlaps it is difficult to get an overview of non-relational databases. A few categories are listed below:

Document	Apache CouchDB, Couchbase, MarkLogic MongoDB
Column	Cassandra, Accumulo, Druid, HBase
Graph	InfiniteGraph, Allegro, Neo4J, OrientDB, Virtuoso, Stardog
Key-Value	MemcacheDB, Redis, Dynamo, Riak FoundationDB, Aerospike FairCom, c-treeACE

Programming Python - CouchDB

Apache CouchDB is one of the most popular open source databases, widely used as a document-oriented NoSQL database. Like most other NoSQL databases, CouchDB uses the JSON (JavaScript Object Notation) format to store data. The JSON format is the open standard data file format that is used as an alternate to XML to transmit data between multiple incompatible and heterogeneous servers.

The following example shows a sample JSON representation:

```
{
  "FirstName": "X",
  "LastName": "X",
  "LiveStatus": X,
  "Gender": X,
```

```
},
"ContactDetails": [
  {
    "ContactType1": "X",
    "Number": "X"
  },
  {
    "ContactType2": "X",
    "Number": "X"
  }
],
"JobStatus": [X],
"PAN": X
}
```

Installing CouchDB

On Ubuntu and Debian Linux systems, use the following command:

```
$ sudo aptitude install couchdb
```

For Gentoo Linux, use the command below:

```
$ sudo emerge couchdb
```

The services can be started or stopped using the *init* scripts in all the distributions as follows:

```
$ /etc/init.d/couchdb start
```

The CouchDB installer is available for Windows from <http://couchdb.apache.org>. The CouchDB installed on the system can be executed in standalone mode as well as in service mode.

Futon: A GUI administrator panel for CouchDB

Futon is the Web based GUI panel that is built for CouchDB. It provides the basic interface for a majority of the functions including creating, deleting, updating and viewing documents. It provides access to the configuration parameters and an interface for initiating replication (see Figures 3 and 4).

CouchDB's interaction with Python

To interface Python with CouchDB, a specialised package called *couchdb* is used with the following main modules:

- *couchdb.client*: This is a client library for interfacing with CouchDB
- *couchdb.mapping*: This module provides the advanced mapping between JSON documents of CouchDB and Python objects
- *couchdb.view*: This module implements the view based server for the views written in Python

```
>>> import couchdb
```

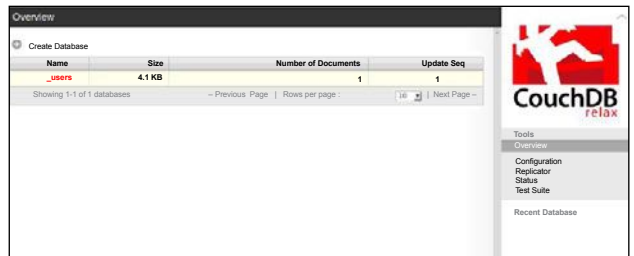


Figure 3: Futon: The administration panel for CouchDB

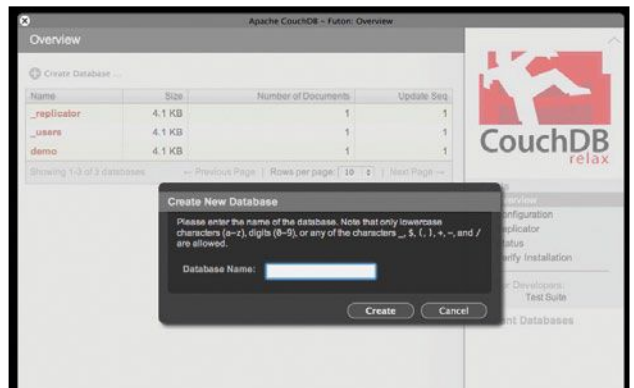


Figure 4: Database creation in Futon

```
>>> couch = couchdb.Server()
```

The above code creates the server object. The default URL is *localhost:5984*.

On live servers, the code will be as follows:

```
>>> couch = couchdb.Server('http://www.mybigcloudportal.com:5984/')
>>> mydb = couch.create('testDB')
```

The above code creates a new database.

```
>>> mydb = couch['mynosqlldb']
```

The above code is to use the existing database.

```
>>> mydoc = {'Country': 'India'}
```

After selecting a database, the above code is used to create a document and insert it into the database.

```
>>> mydb.save(mydoc)
```

The *save()* method shown above returns ID and the 'rev' for the currently created document.

```
>>> mydb.delete(doc)
```

```
>>> couch.delete('testDB')
```


The above command is for cleaning the document and database.

Using CouchDBKit

The goal of CouchDBKit is to provide a dedicated framework for the Python application to manage and access CouchDB.

The following features are inherent with CouchDBKit:

- To use the *http* backend using *py-restclient*
- Managing documents dynamically
- Threadsafe
- Attaching design docs with the application and sending these to CouchDB
- Managing documents with a dynamic schema

In order to install CouchDBKit using Pip, use the following commands:

```
$ curl -O http://python-distribute.org/distribute_setup.py
$ sudo python distribute_setup.py
$ easy_install pip
```

For installation or upgrading to the latest released version of CouchDBKit, use the command given below:

```
$ pip install couchdbkit
```

The following is the code you can use to work with CouchDBKit:

```
from couchdbkit import Server
myserver = Server()
db = myserver.create_db("couchdbkit_test")
db['myid'] = { 'x': 'Hello' }
doc = db['myid']
```

You can map a CouchDB object to a Python object easily with a dynamic schema, as follows:

```
from couchdbkit import Document
class MyDBClass(Document):
    author = StringProperty()
    content = StringProperty()
    date = DateTimeProperty()
MyDBClass = MyDBClass()
MyDBClass.author = "AuthorName"
MyDBClass.homepage = "http://couchdbkit.org"
```

Once this is done, the first CouchDB document will be as shown below:

```
import datetime
from couchdbkit import *
class MyDBClass(Document):
    author = StringProperty()
    content = StringProperty()
```

```
date = DateTimeProperty()
```

Here is the code to save a MyDBClass on the 'MyDBClass' database. We also see how to create a database.

```
server = Server()
db = server.get_or_create_db("MyDBClass")
MyDBClass.set_db(db)
MyDBClass = MyDBClass
(
    author="AuthorName",
    content="Welcome",
    date=datetime.datetime.utcnow()
) MyDBClass.save()
```

Programming Python – MongoDB

First, MongoDB and PyMongo are installed so that the connection of Python with MongoDB can be established. In the Python shell, the following instruction is executed:

```
$ import pymongo
```

Next, the MongoClient is created by running the *mongod* instance. The following code is connected on the default host and port:

```
>>> from pymongo import MongoClient
>>> myclient = MongoClient()
```

The specific host and port can be mentioned explicitly as follows:

```
>>> myclient = MongoClient('localhost', 27017)
```

Alternatively, the MongoDB URI format can be used as shown below:

```
>>> myclient = MongoClient('mongodb://localhost:27017')
```

The MongoDB instance is able to support multiple and independent databases, which can be obtained with the use of attribute style access on MongoClient instances, as follows:

```
>>> mydb = myclient.MyTestDatabase
```

Similarly, other instructions of Python for MongoDB can be used for unstructured and Big Data processing. 

By: Dr Gaurav Kumar

The author is the MD of Magma Research and Consultancy Pvt Ltd. He is associated with a number of academic institutes, where he delivers lectures and conducts technical workshops on the latest technologies and tools. He can be contacted at kumargaurav.in@gmail.com

Go: The Simple Programming Language

This is an introduction to Go, an open source programming language that makes it easy for developers to build simple, reliable and efficient software. The latest stable version of Go, the sixth major release so far, is version 1.5, which was released on August 19, 2015.

The Go programming language was developed at Google by Robert Griesemer and UNIX luminaries Rob Pike and Ken Thompson. It is an open source, high-level, compiled, highly efficient, strongly statically typed programming language. Go supports both procedural and object-oriented programming, though its object-orientation is quite different from other object-oriented programming languages like C++, Java, etc.

Some of the features of Go that we will look at are: object-orientation without class, inheritance and overloading, duck typing, closure, concurrency; built-in support for slices, maps and strings; and fast compilation, among others. Let's install Go first. I will describe the installation steps on Ubuntu 15.04, though it can be installed on any distribution of your choice. Go can be installed either from the binary as follows:

```
$ sudo apt-get install golang
```

...or from the Go source code. I will cover the second method of installation, step-by-step.

Step 1: Install the prerequisites package. We need Git to complete the installation.

```
$ sudo apt-get install git
```

Step 2: Download the Go set-up tree. The Go source code is written in Go itself; so to create the bootstrap, we need a workable Go set-up. Download the Go 1.4 set-up tree and configure it as follows:

```
$ wget https://storage.googleapis.com/golang/go1.4.linux-386.tar.gz
$ mkdir $HOME/bootstrap
$ tar -C $HOME/bootstrap -xzf go1.4.linux-386.tar.gz
$ export GOROOT_BOOTSTRAP=$HOME/bootstrap/go'
```

Step 3: Get the Go source and compile it as follows:

```
$ mkdir $HOME/source
$ cd $HOME/source
$ git clone https://go.googlecode.com/go
```



```
$ cd go
$ git checkout go1.5
$ cd src
$ ./all.bash
```

This takes some time. If everything goes right, you should get the message 'ALL TESTS PASSED' along with some other responses. Now, add the Go *bin* directory to your PATH environment variable, as follows:

```
$ export PATH=$PATH:$HOME/source/go/bin'
```

Step 4: Check for the Go installation, by using the following command:

```
$ go version
go version go1.5 linux/386
```

Your actual output may vary, depending on your computer architecture.

Step 5: Create the *GOPATH* environment variable, as follows:

```
$ mkdir $HOME/workspace
$ export GOPATH=$HOME/workspace'
```

Step 6: Install the Go tool *godoc*, by issuing the following code:

```
$ go get golang.org/x/tools/cmd/godoc
```

Step 7: Check the *godoc* installation by issuing the following command:

```
$ godoc
usage: godoc package [name ...]
...
```

This displays *godoc help*. Here, it's truncated after the first line.

Step 8: Set the required environment variables for Go development, as follows:

```
$ export PATH=$PATH:$GOPATH/bin
$ export GOBIN=$GOPATH/bin
```

Step 9: Add all the export commands executed so far in the *\$HOME/.bashrc* file to make them permanent. Now, log out and log in to get all the environment variables updated.

Our first Go program

Let's write our first Go program, a simple implementation of a stack data structure of integers, i.e., a 'last in, first out' list of integers. In this version, I have intentionally avoided all the bells and whistles of the Go language in order to help readers first understand the basics. We will update our program gradually to learn different concepts of the Go language.

Go code is organised in packages. A package can span over multiple *.go* files. We have two files in this example—the *stack* package in *\$GOPATH/src/gostack/stack/stack.go*, and the *main* package, which contains the *main()* function, the entry point of our program in *\$GOPATH/src/gostack/gostack.go*. Let's go through our *stack.go* code first.

```
$GOPATH/src/gostack/stack/stack.go
// The code in this file belongs to the 'stack' package
package stack
// Import the required packages, here only 'error'
import "fmt"
// Create a 'Stack' type, which is a slice of integers
type Stack []int
// 'Push' function to push item onto the stack given as
argument
func Push(stack *Stack, item int) {
    *stack=append(*stack,item)
}
// 'Pop' function to pop item from the stack given as
argument
func Pop(stack *Stack,item *int) bool {
    if len(*stack)==0 { // len() returns the no of elements
in a slice
        fmt.Println("Pop:: The stack is empty")
        return false
    }
    // remove the last element from the slice in item
```

```
    *item=(*stack)[len(*stack)-1]
    // creates a new slice by taking the ele from first to
second last
    *stack=(*stack)[:len(*stack)-1]
    return true
}
// 'Show' function to display the content of the stack
func Show(stack Stack) {
    if len(stack)!=0 {
        fmt.Printf("[ ")
        for _,item := range (stack) {
            fmt.Printf("%v ",item)
        }
        fmt.Printf("] ")
    } else {
        fmt.Println("Show:: The stack is empty")
    }
}
```

The first *go* statement states the package name 'Stack', which our code belongs to. Then we need to import a set of built-in or custom packages that are needed in our program; here, only the *fmt* package, required for formatted I/O, has been imported. Now, we have created a custom type 'Stack' which is equal to a slice of integers. We have written three functions to operate on a stack -- *Push()*, *Pop()* and *Show()*. The *Push()* function takes a *Stack* pointer and an integer item to push onto the stack. Since a slice, unlike an array, can grow indefinitely, we skipped the overflow check. The *Pop()* function takes a *Stack* pointer and an integer pointer item, where the popped item is returned. It returns a Boolean value to indicate the success or failure of the operation. The *Show()* function takes a stack of value type. It prints the values present in the stack to the console.



Note: The *Push()* and *Pop()* functions take the address of the stack to be operated on, because they need to modify the original stack. But, the *Show()* function doesn't modify the original stack, so it can operate on a copy of it and it is passed as a value. For custom type of a bigger size, it is efficient to use call-by-address, even for 'read only' use cases to avoid copying of a big chunk of memory.

Now let's see our main package, shown below:

```
$GOPATH/src/gostack/gostack.go
// The code in this file belongs to the 'main' package
package main

// import our 'stack' and other required packages
import (
    "gostack/stack"
    "fmt"
)
```



```
// The 'main' function, entry point of our program
func main() {
    var myStack stack.Stack
    var item int
    // show the empty stack
    stack.Show(myStack)

    // try to pop from the empty stack
    if stack.Pop(&myStack,&item) {
        fmt.Printf("\nThe Popped item=%d\n",item)
    }

    // push 10,20 into the stack
    stack.Push(&myStack,10)
    stack.Push(&myStack,20)

    // show the stack
    stack.Show(myStack)

    // pop from the stack
    if stack.Pop(&myStack,&item) {
        fmt.Printf("\nThe Popped item=%d\n",item)
    }

    // show the stack
    stack.Show(myStack)
}
```

The first line states the package name 'main'. Then we import two packages, previously written 'stack' and built-in 'fmt'. Note that the custom package's 'stack' name is mentioned relative to the `$GOPATH/src` directory and it's the directory name of the package source, not the file name. First, we create a variable `stack` of 'stack' type from the `stack` package. It takes the form of: `<package-name>.<type-name>`. One more variable `item` of the `int` type created for `Pop()` call. The rest of the code is self-explanatory. Now, let's compile and execute the program, as follows:

```
$ cd $GOPATH/src/gostack
$ go build
$ ./gostack
Show:: The stack is empty
Pop:: The stack is empty
[ 10 20 ]
The Popped item=20
[ 10 ]
```

We don't have class or inheritance in Go. We have methods and interfaces. Let's understand methods first. A method is a function associated with a particular type. We have changed our functions to methods by shifting the stack reference from the argument list to Go receiver notation just after the `func` keyword. So, the set of three methods `Push()`, `Pop()` and

`Show()`, forms an interface. Any type having the same interface, i.e., set of methods with the same signature as these three, is type compatible in Go. An empty interface is represented by `interface{}`, which is satisfied by any type. That's why we have changed our item type to `interface{}` from `int` so that our stack can handle any type of data this time. Now, we can treat our stack as an object rather than normal variable, and call the operations as methods rather than simple functions. In Go, multiple values can be returned from a function. The `Pop()` function is modified to take advantage of that. In case of a failure, apart from the popped item, we also return an error.

Now, let's modify the program to make it object-oriented in the Go way.

```
$GOPATH/src/gostack/stack/stack.go
// The code in this file belongs to the 'stack' package
package stack
import (
    "errors" // import the 'error' package for error
             reporting
    "fmt"     // import the 'fmt' package for displaying
             output
)
// Create a 'Stack' type, which is a slice of interface{}
type Stack []interface{}

// 'Push' function to push item onto the stack given as
receiver
func (stack *Stack) Push(item interface{}) {
    *stack=append(*stack,item)
}
// 'Pop' function to pop item from the stack given as
receiver
func (stack *Stack) Pop() (interface{},error) {
    if len(*stack)==0 { // len() returns the no of elements
in a slice
        return nil,errors.New("Pop:: The stack is empty")
    }
    // remove the last element from the slice in item
    item:=(*stack)[len(*stack)-1]
    // creates a new slice by taking the ele from first to
second last
    *stack=(*stack)[:len(*stack)-1]

    return item,nil
}
// 'Show' function to display the content of the stack given
as receiver
func (stack Stack) Show() error {
    if len(stack)!=0 {
        fmt.Printf("[ ")
        for _,item := range (stack) {
            fmt.Printf("%v ",item)
        }
    }
}
```

```

    fmt.Printf("] ")
    return nil;
} else {
    return errors.New("Show:: The stack is empty")
}
}

```

We have changed our *main* accordingly. This time, we have used multiple assignment notations to copy the data as well as error value returned from *Pop()*.

```

$GOPATH/src/gostack/gostack.go
// the code in this file belongs to the 'main' package
package main
// import our 'stack' and other required packages
import (
    "gostack/stack"
    "fmt"
)
// the 'main' function, entry point of our program
func main() {
    var myStack stack.Stack

    // show the empty stack
    if myStack.Show()!=nil {
        fmt.Println("Show:: The stack is empty")
    }
    // try to pop from the empty stack
    if item,err:=myStack.Pop(); err==nil {
        fmt.Printf("\nThe Popped item=%v\n",item)
    }else{
        fmt.Println("Pop:: The stack is empty")
    }
    // push 10,20 into the stack
    myStack.Push(10)
    myStack.Push(20)
    // show the stack
    myStack.Show()

    // pop from the stack
    if item,err:=myStack.Pop(); err==nil {
        fmt.Printf("\nThe Popped item=%v\n",item)
    }else{
        fmt.Println("Pop:: The stack is empty")
    }
    // show the stack
    if myStack.Show()!=nil {
        fmt.Println("Show:: The stack is empty")
    }
}

```

Now, it's time to add exception handling to our program. We have modified our *Pop()* method to demonstrate how exception handling works. We have

panic() and *recover()* built-in functions. A detailed description of these functions is beyond the scope of this article. We will only describe our own use case here. If *Pop()* method is called on an empty stack, we generate a panic with a user-given message. By default, *panic* gets propagated through the call stack of functions. But, we want to capture it in the same function and handle it. At the beginning of the *Pop()* we have called an anonymous function, i.e., a function without name. But this doesn't get executed immediately because of the *defer* keyword. This is called a deferred function, which gets executed at the end of the function before returning to the caller. That means that just before leaving the *Pop()* function, we check if any panic has been generated, and if so, we print the panic message to the console.

```

$GOPATH/src/gostack/stack/stack.go
// 'Pop' function to pop item from the stack given as receiver
func (stack *Stack) Pop() (item interface{},e error) {
    e=errors.New("Empty stack")
    defer func() {
        if ret:=recover(); ret!=nil {
            fmt.Printf("%v\n",ret)
        }
    }()

    if len(*stack)==0 { // len() returns the no of elements
in a slice
        // call panic() with the error msg
        panic("Pop:: The stack is empty")
    }
    // remove the last element from the slice in item
    item=(*stack)[len(*stack)-1]
    // creates a new slice by taking the ele from first to
second last
    *stack=(*stack)[:len(*stack)-1]
    return item,nil
}

```

In the next article on Go, we will discuss concurrency support along with a suite to test a Go program. **END** 

References

- [1] 'An Introduction To Programming in Go' by Caleb Doxsey
- [2] 'Programming in Go' by Mark Summerfield
- [3] <https://golang.org>

By: Ramaprasad Chakraborty

The author works at ISRO, Sriharikota, as a scientist engineer SD and has six years of industry experience. He is interested in real-time operating systems, controller redundancy mechanisms, networking, cyber forensics, database management systems and task automation with different levels of scripting and various open source technologies.



Continuing with our series on building Android apps with App Inventor 2, we present our DIY fans with a project to build a simple music player app with three buttons.

In the last issue of *OSFY*, we developed an SMS assistant application using App Inventor 2. Let's now create a music player app. Listening to music on an Android phone is one of the greatest pleasures. Android phones normally have a music player app installed, by default, apart from which there are many custom made music players available in Google's Play Store. This tutorial shows you how to create a simple music player application using App Inventor 2.

The first step in the process will be to set up a GUI that will make it easy to assign or implement certain functions in the application.

GUI requirements

We will need three buttons—the *Play*, *Pause* and *Stop* buttons to accomplish the simple tasks of playing the music, pausing it and stopping it.

For this project, we will require the components shown in Table 1.

Building the player

To build the player, follow these steps:

1. Drag and drop the components mentioned in Table 1, to the viewer.
2. Visible components can be seen by you while the non-

visible components will be located beneath the viewer under the tag 'Non-visible'.

3. There is a label for the name of the application.
4. All buttons need to be put within the 'Horizontal arrangement' so as to keep them aligned horizontally.
5. If you have dragged and placed everything, the layout will resemble the diagram in Figure 1.

Make the necessary property changes like we did in the previous article to change the text property for the label and button components.

Since we need to play music with the help of the player, upload the media file to the server. I hope you know how to do this.

Now your graphical user interface (GUI) is ready. This is exactly how the application will look after the installation.

We next head towards the block editor to define the GUI's behaviour. So let's discuss the actual functionality that we expect from our application.

1. We need to code it so that on pressing the *Start* button, it should start playing sound.
2. On pressing the *Pause* button, it should pause the audio play, which can be resumed from the same place on pressing the *Play* button again.
3. On pressing the *Stop* button it should stop the audio completely.

Table 1

	Component name	Purpose	Location
1	Horizontal Arrangement	To arrange the child components horizontally	Palette--> Layout--> Horizontal Arrangement
2	Label	To display a label	Palette-->User Interface-->Label
3	Button	One button to serve as <i>play</i>	Palette-->User Interface-->Button
4	Button	One button to serve as <i>pause</i>	Palette-->User Interface-->Button
5	Button	One button to serve as <i>stop</i>	Palette-->User Interface-->Button
6	Player	To play the audio	Palette-->Media-->Player

So let's move on and add these functionalities using the block editor. The earlier article in this series discussed how to switch from the designer to the block editor. However, just to remind you, there is a button available right above the *Properties* pane to switch between the designer and block editor.

The block editor

I have already prepared the blocks for you. All you need to do is drag the relevant blocks from the left side palette and drop them on the viewer. Arrange the blocks the same way you see them in the image. I will explain what each block does and how it is called.

The blocks shown in Figure 3 are easy to understand. They just say 'play', 'pause' and 'stop' the player when the particular button is pressed. And that's it!

Now we are done with the block editor too. Next, we will move to downloading and installing the app on your phone to check how it is working.

Packaging and testing

To test the app, you need to get it on your phone. First, you have to download the application to your computer and then move it to your phone via Bluetooth or a USB cable. I'll tell you how to download the app.

On the top row, click on the *Build* button. It will give you the option of downloading the *apk* to your computer.

You can view the progress of the download and after it is successfully completed, the application will be placed in the *download* folder of your directory or in the location you have selected for your download.

Now you need to get this *apk* file to your mobile phone either via Bluetooth or via USB cable. Once you have placed the *apk* file on your SD card, you need to install it. Follow the on-screen instructions. You might get some notification or warning regarding installation from an untrusted source. Allow this from the *Settings* and after successful installation, you will see the icon of your application in the menu of your mobile.

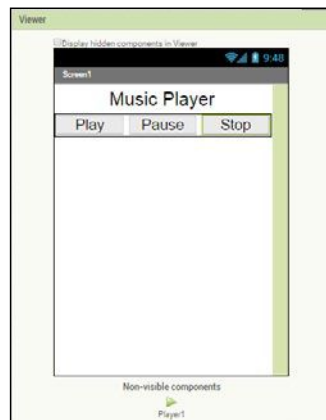


Figure 1: The viewer



Figure 2: The components

Here you will see the default icon, which can be changed.

I hope your application is working exactly as per your design requirements. Now, depending upon your choice, you can customise various things like the image, sound and behaviour too.



Figure 3: Block editor

Debugging the application

We have just created the prototype of the application with very basic functionality. Now come the various cases that require serious attention so as not to annoy the user, and which your app should be able to sustain. Consider the cases below:

1. The app is coded to play a single audio file which the developer has uploaded to the server. Can it also access media files from local storage, like the SD card?
2. We have labelled the buttons with simple text. Can we, instead, put images over the buttons as symbols for *play*, *pause* and *stop*?
3. Can we add visuals or media information during the play?
4. Can we show how much of the music has been played in terms of duration (minutes or seconds)?

These are some of the scenarios that could possibly occur, and which the user would be pretty happy to see implemented. Think about how you can integrate these functions into the application. Do ask me if you fail to accomplish any of the above cases.

You have successfully built another useful Android app for yourself—a music player with three button controls. Happy inventing! **END** 

By: Meghraj Singh Beniwal

The author is a B. Tech in electronics and communication, a freelance writer and an Android apps developer. He is currently working as a systems engineer at Infosys, Mysore. He can be contacted at meghraj Singh01@rediffmail.com

Getting Started with Dart



Dart is an open source Web programming language that has been rolled out by Google. It is scalable, and has robust libraries and runtimes for building Web, server and mobile apps. Touted as a rival to JavaScript, it was created out of the frustrations a few developers had with some aspects of the 20-year old language. Dart holds great promise for the future.

Dart is an open source, scalable programming language, maintained by Google. It's relatively easy-to-learn and can be used for building Web, server and mobile applications. It's not only just a new programming language but also a powerful new platform for modern Web development.

Dart enforces a structured design for programs, which other client-side programming languages like JavaScript don't do. Dart reduces most of the problematic issues with JavaScript such as the tendency towards sloppy, hard-to-maintain code.

JavaScript doesn't come with features like class hierarchy and inheritance, which makes the programming of large projects difficult. Dart solves this issue. Native Dart is twice as fast as JavaScript and can also be compiled back to JavaScript for any modern browser to run. It comes with a huge collection of built-in libraries and a packaging system. Using Dart, you can quickly write prototypes that evolve rapidly, and you also have access to advanced tools, reliable libraries and good software engineering techniques.

Getting started

To get started with learning Dart, <http://www.dartlang.org> provides an interactive Dartboard, which lets us write and run Dart code in the browser itself. The website also offers the Dart Editor, a downloadable text editor which lets us create, modify and run Dart apps on local machines. An SDK, which includes various command line tools and the Dart VM, can also be downloaded.

If the browser doesn't support an embedded Dart VM, you can use the built-in Dart-to-JavaScript compiler that comes with the Dart SDK to compile Dart code into JavaScript.

Dart is designed for mass adoption, so it has to have a familiar feel for both scripting language users (such as JavaScript users) and structured language users (such as Java developers).

Let's now write a simple Dart program.

Don't forget to go through the programming paradigms of Dart at <https://www.dartlang.org/docs/dart-up-and-running/ch02.html#important-concepts>.



Figure 1: Dart logo

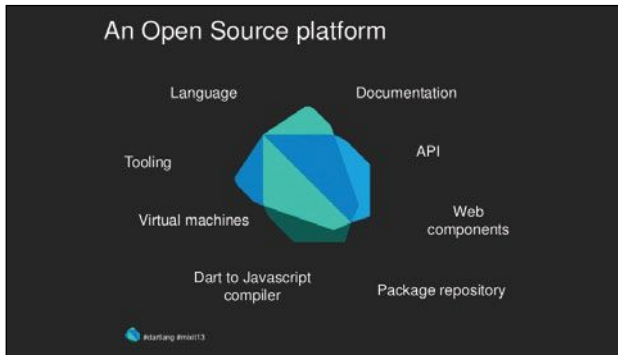


Figure 2: Dart: An open source platform

Hello, Dart!

Let's start writing a simple program in Dart by using the following commands. The syntax is very similar to any C based language.

```
//SimpleDart.dart
void main() {
  print("Hello, World!");
}
```

One of the design goals of Dart is a simple and familiar syntax. This application may be very basic, but it gives us some information about how Dart is written. First off, we can see that it uses a C-style syntax. Second, rather than JavaScript being written just about anywhere you want, our Dart code lives within a main function. That provides consistency to the code.

Although the 'Hello world' script looks simple, Dart is technologically more advanced than JavaScript and since it aims to fix security and other problems of JavaScript, it definitely has an appeal.

Now let's move on and write a more advanced Dart program.

Your first 'real' Dart program

The first thing you need to do is to set up the development environment for Dart, for which you have to install Dart Editor. It comes with the SDK and all the tools you need to get started. You can also configure any other IDE to support Dart by installing the required plugins.

We are going to keep this simple, but rather than using the command line, let's say hello to the browser. Let's create an HTML element that, when clicked, produces an alert with the

text, 'Hello, World!'

First, open Dart Editor and click on *File -> New Application* and name it *SimpleDart*. Make sure you select the command line application, and leave *Generate sample content* selected.

In your SimpleDart project, open the file called *pubspec.yaml* and select *Source* at the bottom of the editor. You need to add a dependency called *browser*. Here's how it looks:

```
name: SimpleDart
description: A simple Dart program.
dependencies:
  browser: any
```

Now, create an HTML file that you can use for your application's display. Right click the bin directory in your *Project navigator*, and select *New File*. Name the file *SimpleDart.html*.

Once the file is created, you should see a very basic HTML file populated with some standard elements, as shown below:

```
<p id="text"></p>
<script type="application/dart" src="SimpleDart.dart"></script>
```

The paragraph tag with the ID of the text gives you a placeholder, where you can put some text from your Dart code. Second, the script tag here is pointing to SimpleDart, and has a type of application/dart. This instructs the browser to load your Dart code and execute it from its Dart VM.

Now that you have an HTML page for your display, modify your *SimpleDart.dart* file to put some text in the paragraph tag. Before you can access the HTML though, import the appropriate library to give you HTML support, as follows:

```
//SimpleDart.dart
import 'dart:html';
```

Now create libraries that other applications can import. You are finally ready to write some code. Use a Dart method called *query*, which will search for an HTML element and allow you to modify it programmatically. Let's search for *#text*, which will give you access to the paragraph tag we saw above. Within the main function, use the following command:

```
query('#text').text = "Hey, Browser!";
```

Go ahead and run this. You will notice that the Chromium browser opens up automatically, and shows you a blank page with the text, 'Hey, Browser!' printed on it.

To add a click-event, there are a couple of options. You

could either query for the element again, store the element in a variable, or chain the commands.

```
//SimpleDart.dart
import 'dart:html';

void main() {
  query('#text')
    ..text = "Hey, Browser!"
    ..onClick.listen(handleClick);
}

void handleClick(MouseEvent event) {
  print("Open Source For You.");
}
```

Rather than using a single dot (.), use two dots (..) to indicate that the commands are going to be chained. Only after the final command do we enter a semi-colon to indicate that the chain has been completed.

You will also notice that we have defined a new method that accepts a *MouseEvent* parameter. When the *#text* element is clicked, it will invoke *handleClick* and pass with it the click-event that caused it.

Go ahead and run this, and when you click on 'Hey, Browser!' you should notice in your IDE's console that the text, 'Open Source For You' has been printed. It's good to know that even in Web application mode, we can still log to the command line. This will be great for debugging applications down the line.

Now let's wrap up this application. Inside the *handleClick* event, create an alert dialogue box with the text, 'Hello, World!' by using the following code:

```
void handleClick(MouseEvent event) {
  window.alert("Hello, World!");
}
```

Run the application, click the text again, and you should

see a pop-up dialogue box with the text, 'Hello, World!'

Dart for servers

Apart from developing Web applications, Dart, like JavaScript, can also be used to code the servers. Dart VM server runs everywhere —from a Linux, Windows or a Mac prompt. It has been ported to i32, x64, ARM, ARM64, and MIPS. Asynchronous code is easy to write, with the new *async/await/yield* language features. Spawn an isolate to run Dart functions and libraries in an *isolated heap*, and take advantage of multiple CPU cores. Connect to your VM through your browser, and get real-time insights into your running app. You can also access files, UDP and TCP sockets, HTTP, WebSockets, and more. You can delve deep into <https://www.dartlang.org/codelabs/server/> to learn more.

Dart for mobiles

The ability to write a Dart app for deployment on a variety of mobile devices is under development. Fletch, a Dart runtime that's currently under development, aims to support app deployment on Android and iOS. Future versions of the Dart SDK will include Fletch.

What next?

What we've done here is developed a very simple Web application in Dart. There is a Dart Web UI library, which can be used to do templating and data binding to simplify our Dart code even further. If you are looking for a more in-depth coverage of the Dart language and want to see how to build a real application with Dart, check out <https://www.dartlang.org/docs/tutorials/>.

To contribute to the Dart programming language or to stay updated on the latest Dart news, the official mailing list is at <http://www.dartlang.org/mailling-list>.  **END**

By: Srirachan Chiruvolu

The author is a FOSS enthusiast and an aspiring computer science engineer. He loves to explore new open source technologies. You can reach him via email at mail@srirachan.xyz.

Read more stories on LED at
www.electronicb2b.com

**TOP
LED STORIES**

- The latest in LED indoor displays
- The latest in LED street lights
- The latest in LED office lights
- Learn how to become a channel partner
- Read about brand strategies

**ELECTRONICS
B2B**
INDUSTRY IS AT A

CLICK OF A BUTTON

Log on to www.electronicb2b.com and be in touch with the Electronics B2B Fraternity 24x7



JavaScript:

The New Parts

ECMAScript2015 (ES6) is the specification that defines new features in JavaScript. In the second part of this series on JavaScript, let us look at new features like the binary/octal literals and enhanced object literals.

In the first part of this series, we covered the direction in which JavaScript (JS) is evolving. We also briefly looked at the new themes in the ECMAScript2015 specifications, before going on to cover block scope and template strings. And we explored two environments where you could try the new version—*es6fiddle.net* and *transpilers*.

In this part, we will cover more JS environments and look at the level of coverage in each of them. The browser environment has a ready to use debug environment.

Table 1 lists the level of coverage for features explained in the earlier article.

Table 1
(None: 0 per cent; Low: > 0-30 per cent; Medium: 30-70 per cent; High: 70-95 per cent; Full: 100 per cent)

	Non-browser environments			Browsers (stable version)	
Feature	io.js 3.0.0	Babel	node.js 0.12.0	Chrome 44	Firefox 40
Block scope	High	High	Low	High	Low
Template strings	Full	High	None	Full	Full

Language environments and their support

io.js: io.js is a famous fork of node.js that was created with the intention of having community control over the future of the compatible node.js environment. io.js has far more ES6 feature support than node.js. Learning new JavaScript features could be a compelling reason for you to move from node.js to io.js, which is a JS parser with a command line from which you can directly try ES6 features. Setting up io.js is covered in the next section.

Babel: This is more than a translator for ES6-to-ES5 conversion. Babel is recommended to JS enthusiasts who love

to understand the nuances of language better. You can see how ES6 features can be implemented with ES5. If you are keen to learn about compilers, you can do so by dissecting Babel, which has been developed by a JS enthusiast with the intention to learn JavaScript!

Node.js: Node.js support for ES6 features is limited. Please note that in some cases, you may need to enable a `--harmony*` flag to get the required feature support. For simplicity of representation, the need for flag-enabling is not explicitly mentioned in the table.

In browser environments, though the current version of Chrome seems to support ES6 better (w.r.t. features covered in Parts 1 and 2 of this series), the beta and nightly versions of Firefox have better overall ES6 support than Google Chrome. If you are a Google Chrome fan, consider switching to Firefox to experiment with ES6 from within browser environments.

Setting up io.js

Visit <https://iojs.org/en/index.html> and download the bundle. In the Linux (Ubuntu) environment, io.js is just a Zip file to be downloaded and unzipped. You can create a soft link to the node.js file, if you would like to have a single JS environment. io.js claims maximum compatibility with node.js. But if your intention to try io.js is only ES6, you can use io.js separately. Begin by issuing the following code:

```
janardan@ubuntu:~$ iojs --version
v3.1.0
janardan@ubuntu:~$ iojs
> greet = "Hello, io.js";
'Hello, io.js'
> console.log(`Message: ${greet}`);
Message: Hello, io.js
```


From the command line, I tested the template string feature. To exit from the command line, type `.exit` (dot exit).

Here's a sample snippet of how node.js will behave for the same input.

```
janardan@ubuntu:~$ node --version
v0.12.6
janardan@ubuntu:~$ node
> greet = "Hello, node.js";
'Hello, node.js'
> console.log(`Message: ${greet}`);
...

```

In the last statement, as node.js does not understand *backtick*, the node continues to expect input from the user by giving ellipses.

This month, I cover topics that are really additions to existing ES5, rather than entirely new features.

The literals are supported as part of the theme, where JS should go low-level (as in close to the machine and bits). This is useful for programmers doing bit manipulations, such as file permissions or network packet processing.

The array, string and math library is to eliminate the need for programmers to use other libraries (such as *Underscore.js*), or to code their own helper functions.

Binary and octal literals

There are two new number literals that have been added to JavaScript. These are the binary and octal literals. This is actually a re-introduction from previous releases.

The *0b* (zero-b) prefix is for the binary numbers and *0o* (zero-oh) prefix is for octal numbers.

```
> 0b111 // Output: 7 (decimal)
```

The output is in decimal.

```
> 0b11+0o11 // Output: 12 (3+9)
```

The following is the sum of the binary literal and the octal literal. The result is decimal.

```
> 0b11+0o11+0x11+011 // Output: 38
```

The following is the sum illustrating binary, octal, hex and octal. Note that JS interpreters still support *0* (zero) prefix notation for octal.

```
> "use strict"; 0b11+0o11+0x11+011
SyntaxError: Octal literals are not allowed in strict mode.
```

When you run the same in *use strict* mode, you get an error as only the *0* (zero) prefix octal is deprecated.

Enhanced object literals

Enhancements to object literals involve a cluster of features rather than a single one. The first is shorthand for object properties at the time of initialisation. Shown below is an example that best explains it. The second enhancement is the shorthand for object methods, and the third is computed property names.

Properties shorthand

In ES5, when we constructed object properties with scalar variables, we repeated the variable names like what's shown below in Snippet 1:

```
let firstname = "Janardan",
    lastname = "Revuru",
    organization = "HP EG India R&D";

let employee = {
  firstname: firstname,
  lastname: lastname,
  organization: organization,
};
```

Snippet 1: Longform notation

In ES6, we can avoid the repetition of object properties and variables names, if they are the same, as shown below in Snippet 2:

```
let employee = {
  firstname,
  lastname,
  organization,
};
```

Snippet 2: Shorthand notation

You can even mix shorthand notation with direct initialisation, like what's shown below in Snippet 3:

```
let employee = {
  firstname,
  lastname,
  org: organization,
  location: "Mahadevapura"
};
```

Snippet 3: Mixed notation

Method shorthand

ES6 also provides shorthand notation for the method definitions. You can eliminate the keyword 'function' when defining method properties.

Extending Snippet 2, we can write the code as shown

below. You will notice that you eliminated writing the *print*: *function { }* to define a method.

```
let payroll = {
  _employees: [],
  print() {
    console.log(JSON.stringify(employee));
  }
};
payroll.print();
```

Output:

```
{"firstname":"Janardan","lastname":"Revuru","organization":"HP  
EG India R&D"}
```

The full code snippet is available at <http://www.es6fiddle.net/iedsjdz/>

Table 2
(None: 0 per cent; Full: 100 per cent)

	Non-browser environment			Browsers (stable version)	
Feature	io.js 3.0.0	Babel	node.js 0.12.0	Chrome 44	Firefox 40
Binary, octal literals	Full	Full	Full	Full	Full
Enhanced object literals	Full	Full	None	Full	Full

Why learn JavaScript?

In the crowded space in which new programming languages emerge every year, with each 'selling' itself as the best ever in the history of computing, how can programmers decide or choose between them?

Over the years, what I have learnt is, it is not the most elegant and feature-rich language that survives the test of time. For a language to continue being in use, there are many factors that come into play. JavaScript has stood the test of time, to some extent, on the basis of its own merits and partly due to luck.

JavaScript does have its share of imperfections. But the sheer omnipresent nature of the JavaScript environment inside browsers on different platforms makes it the most used. With ES6, there are enough reasons to make JavaScript more compiler and translator friendly. This will enable the bulk of programs already written in languages like C/C++ to be translated to JavaScript to run in a browser environment. Some examples are the GWT compiler that translates Java source to JavaScript source, and Sharp# which converts the JavaScript environment. For a list of languages that compile to JavaScript, visit <https://github.com/jashkenas/coffeescript/wiki/List-of-languages-that-compile-to-JS>. So even if you are a non-JavaScript programmer, who wants your application to run in a browser environment, it is essential you understand the fundamental capabilities and limitations of the language.

If you are a graduate out of college or in the early years of your software career, you can bet your career on JavaScript—as some sort of 'one language that fits all purposes', be it developing hobby mobile apps or enterprise class applications. JavaScript has made inroads into the server side too. Thanks to node.js and JSON data interchange, now server side programmers and client side programmers talk the same language! They also exchange data in common JSON format.

To know more about the layers JavaScript is striving, log on to the URL mentioned in Reference 3.

Next month we will cover computed property keys and String functions in ES6. 

References

- [1] 'Play around with ES6 features online'; <http://www.es6fiddle.net/>
- [2] 'Languages that compile to JavaScript' <https://github.com/jashkenas/coffeescript/wiki/List-of-languages-that-compile-to-JS>
- [3] 'An Introduction to Full-Stack JavaScript'; <http://www.smashingmagazine.com/2013/11/introduction-to-full-stack-javascript/>

By: Janardan Revuru

The author works at HP EG India R&D in Bengaluru. He is passionate about JavaScript and related Web frameworks.

THE COMPLETE MAGAZINE ON OPEN SOURCE

OpenSource

ForYou

Your favourite Magazine on Open Source is now on the Web, too.

OpenSourceForU.com

Follow us on Twitter@LinuxForYou

An Easy-to-Follow Git Based Development Workflow

This article provides a cookbook-style Git workflow to help project teams and developers. The workflow is intended for small open source projects that have a few people as maintainers and multiple developers contributing to them.



Git is a powerful distributed version control system that is free and open source. It was created by Linus Torvalds in 2005 when the Linux project had to move away from BitKeeper. Since then, there has been constant addition of functionality and improvement in performance. Today, it is recognised as one of the fastest distributed version control systems. As a result, in the past few years, many open source projects have migrated to Git from SVN, CVS and other version control systems.

However, the power of Git also means that it can be incorrectly used, resulting in confusing revision history after merges and consequent difficulties in tracking code changes. At worst, it could result in loss of code as well. Also, many developers with a background in centralised version control

find it difficult to understand the distributed nature of Git and how to use it effectively. This has created a mental barrier, where project teams hesitate to use Git, due to its perceived difficulty and uncertainty in workflow.

There are many books that describe how to use Git and also cover its internals. But these get into complex details, and do not provide a template workflow that can be used by project teams that are embarking on new development but are new to Git.

Installation and initial configuration

Git is packaged by most Linux distributions. So you can use *apt*, *yum* or any other package management system provided by your distribution to install Git.

However, in case the version provided by your

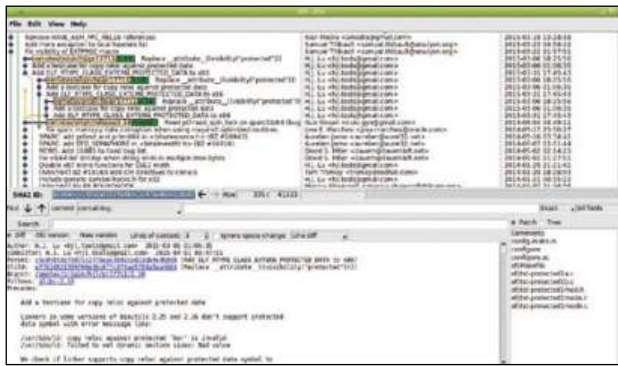
Figure 1: Sample *gitk* output

Figure 2: Shell prompt showing the Git branch

distribution is older and you would like to install the latest version, you can obtain the source from <https://git-scm.com/>. This may be the only option in case you do not have administrator access to install packages, yet want to install Git in your home directory.

Once done, you need to fill in information about yourself in Git so that it can be used in the commits made by you. This step is mandatory for every developer. So proceed as follows:

```
$ git config --global user.name "Your Name"
$ git config --global user.email "your.name@yourcompany.com"
```

Support tools

Along with Git, it is useful to install a graphical tool that displays the history of a Git repository. This tool is called *gitk*. If this is not already on your system, it can be installed via the *gitk* package.

A sample output of the tool is shown in Figure 1.

For bash users, another useful addition is to have Git auto-completion added to your bash shell. This helps you type Git commands faster on the bash command line, by pressing the tab after typing the first few characters of the Git command. The Git auto-completion bash script can be downloaded from <https://github.com/git/git/blob/master/contrib/completion/git-completion.bash>

You can then add the same into your *.bashrc*. For example, the following code can be added to *~/bashrc*:

```
source ~/.git-completion.bash
```

In addition, the PS1 prompt can be modified to show the branch that you are in, when inside a Git repository. The *.bashrc* can modify the PS1 prompt as follows:

```
PS1='[\u: \w$(__git_ps1 " (%s)")]\$ '
```

This results in an output as shown in Figure 2.

This is very handy when you are working on multiple branches. A common error that developers make, i.e., committing code to the wrong branch, can be avoided with this simple step.

Official repository

In Git, all repositories are complete and equal. All developers own their repository. Hence, the official repository just refers to the Git repository identified by the project as 'official' and usually owned by *maintainer*, who is identified by the project team.

All developers clone their individual repositories from this official repository. Note that the *maintainers* themselves may have a cloned repository where they do the development work.

All official builds (release builds, nightly builds, etc) need to be made out of the official repository. Therefore, all code needs to be brought into the official repository ultimately. Thus, code can be considered as committed to the project only after it reaches this official repository from the individual developer's repository.

Cloning the official repository

The official repository can be cloned as follows:

```
$ git clone ssh://userid@ip-addr/path-to-repo [<dir-name>]
```

<dir-name> is optional. If omitted, the same name as in the official repository is created. The Git repository which is cloned is called *origin*.

Branches

Every Git repository has a *master* branch. The *master* branch of a clone is not the same as the *master* branch of the origin. But the *master* branch of a cloned repository tracks the *master* branch of *origin*. It is called a *tracking* branch.

Cloning automatically creates the *master* branch in the cloned repository. Other branches can be created using the following command:

```
$ git branch <branch-name>
```

The branch is local to the repository where it is created. The local branch can be pushed to *origin*.

To move to a specific branch, use the following command:

```
$ git checkout <branch-name>
```

Rules relating to tracking branches

As mentioned above, the *master* branch of a cloned Git repository is a tracking branch. You can track any other

branch in the origin Git repository using the following command:

```
$ git checkout <remote-branch-name>
```

This creates a local branch with the same name as the remote branch, and which tracks the latter.

Now, here are some rules to be adhered to when working on tracking branches:

- Never work directly on any tracking branch
- Create a development branch off a tracking branch for changes that need to go into that tracking branch
- Use `$ git pull --all` in each tracking branch periodically, to keep it in sync with the *origin* repository

You can use the tracking branch as a reference to post code changes for review. For example, if the popular review-board (<https://www.reviewboard.org/>) code review tool is being used by your project team, you could use the following command:

```
$ post-review -tracking-branch=<tracking-branch-name>
```

...to post reviews for changes in that development branch.

The Git workflow

When working on a feature or a defect, it is best to adhere to the steps provided below as your development workflow. The purpose of these steps is to have a single commit addressing a single issue, which is consistent and complete in itself, while also having a clean merge history in the official repository.

At this point, we will assume that your repository reflects the latest state of the official repository. If you have a repository cloned a while ago, pull the latest changes pushed to the official repository, using the following command:

```
$ git pull --all
```

Step 1: Create a branch where you will modify code for the feature/defect assigned to you. Create the branch from the *master* branch, or from a *release* branch, to where the code must ultimately go.

Use the branch only for the specific feature or defect. Do not mix code changes for different work items in the same branch.

Step 2: Modify the code and test it.

Step 3: Commit it to the *working* branch.

Step 4: Send the code for review using the code review tool that your project uses, for example, *review-board*.

Step 5: Rework the code based on review feedback, re-test it and resend for further review. Steps 2 to 5 are repeated until review approval is obtained.

Step 6: Once review approval is received, update the *master* branch and rebase the *working* branch to get the development repository in sync with the latest code. While code changes are merged automatically most of the time, sometimes manual code merges may need to be done.

Step 7: Squash multiple commits (say, due to review feedback) into a single commit. Based on experience, I can say this is seldom done by developers. But, this is important to ensure that the Git version history looks clean and there is a single diff for a single unit of code change. This also makes it easy to cherry-pick a change into another branch.

Step 8: Send the patch to the maintainer of the official repository to incorporate your changes into the main repository.

First, ensure you are in the *master* branch. Now, you can create a branch to make the code changes. Say you are fixing *bug-ID 1234*. Then, the `<branch-name>` below can be *bug1234*.



Note: We will use the *master* branch as an example here, though it could be any other branch in the official repository where you want the code changes to go.

The few specific commands that can be used for each of the above steps are detailed. Comments are shown below in C-style syntax. These are for clarity only and not to be typed.

```
$ git branch <branch-name> /* Create branch */
$ git checkout <branch-name> /* Move to branch */

/* Modify code */
/* Build & test */

$ git add <files> /* Stage file for commit */
$ git commit /* Give meaningful commit log */

/* Send for review */
```

After review approval, update the *master* and rebase your branch. Rebasing is an essential step, which ensures that any changes made by other developers and pushed to the official repository (after your cloning) are visible in your branch. Any merge conflicts can be resolved by you, facilitating a clean merge by the maintainer.

```
$ git checkout master /* Move to master branch */
$ git pull --all /* Sync with latest code in origin's master branch */
$ git checkout <branch-name> /* Move back to working branch */
$ git rebase master /* Sync branch with latest code */
/* Fix merge conflicts, if any, and re-test */
```

Now, squash multiple commits into a single commit via interactive rebase.

```
$ git rebase -i HEAD-<#>
```

In the above command, `<#>` is the number of commits from HEAD to squash; for example:

```
$ git rebase -i HEAD-3
```

Pick only the required commit and squash other commits with the one picked.

The changes can now be submitted to the maintainer for incorporation into the official repository.

```
$ git format-patch origin (or) $ git format-patch -1
```

The above command will generate files of the type `0001-xxxxx.patch`. If the commit squash was done, only one file will be generated. Send these files to the maintainer via email.

What maintainers normally expect is that the mail containing a patch to be incorporated in the official repository should also mention the branch to which the patch is to be applied, the bug-ID, and the proof of review approval (such as the review URL). The maintainer will usually send a confirmation once the patch is applied.

You can now pull changes from the main repository into your *master* branch and delete the *working* branch, using the following code:

```
$ git checkout master
$ git pull -all
/* Verify submitted changes are present in "master" */
$ git branch -d <branch-name>
```

Working on multiple work items

Normally, a developer works on multiple work items. For example, the developer may be assigned a feature and multiple bugs. Having a separate cloned repository for each work item is ideal. But due to disk quota, this is a luxury for most developers. Instead, they need to work on the same cloned repository. This is facilitated by Git, which provides a way to stash work and come back to it later.

As mentioned earlier, each work item must be on a separate branch. So, we can save uncommitted code before switching branches by stashing the changes, as shown below:

```
$ git stash save "<message>"
$ git checkout <other-branch>
...
$ git checkout <earlier-branch>
$ git stash pop --index
```

You may stash changes in multiple branches. To view the stashed changes, use the following code:

```
$ git stash list
```

To apply a specific stash once you are back in a branch, use the command shown below:

```
$ git stash apply <stash-name> --index
$ git stash drop <stash-name>
```

Note that *git stash drop* is used to remove a stashed set of changes after it is applied via *git stash apply*. Git does not remove it automatically, on the assumption that you may need it later for some other branch. This behaviour is unlike that of *git stash pop*, which removes the stashed changes that were popped out.

Labelling

Any project would want to label code at specific milestones. Git allows us to label any specific code snapshot. Apply a label with a comment, using the following code:

```
$ git tag -a <label> -m "<info on label>"
```

To get to a labelled code snapshot, use the following command:

```
$ git checkout <label>
```

Release management

Release management deals with creating a release version of your project for external consumption. Project members label the code snapshot used to create the release version. This helps the project team to go back to the specific code snapshot at any time in the future, often to resolve issues reported in the released product.

The release is usually done from the official repository, where you need to go through the sequence of steps that follow:

- Apply a label with *ver#* as per project conventions, e.g., *myproject_release_v1.0*
- Also, create a branch with *ver#* as per project conventions, e.g., *myproject_release_1.0_branch*
- There are some rules when it comes to *release* branches in the official repository:
 - *Release* branches must never be deleted.
 - *Release* branches must not be rebased with the *master* or any other branch. *Release* branches are long-lived branches, unlike *development* (*working*) branches described earlier.
 - Label any hot-fix releases done out of the *release* branch as per project conventions.

Cherry-picking

When defects are reported in a released version, there could be a need to fix code in a *release* branch and issue updates.

The recommended workflow for handling issues in a *release* branch is as follows:

Step 1: Check if a defect exists in the *master* branch, and if so, fix it in the *master* branch.

Step 2: Cherry-pick the commit into the concerned *release* branch by using the following code:

```
/* Do bug fix and commit to master */
$ git checkout <release-branch>
```

Continued to page 85...

The Rise of **Mobile** Application Development Platforms

There is a huge market for enterprise-ready mobile cross-platform development tools as a result of the increasing diversity in the mobile device ecosystem. Enterprises of all sizes are craving for solutions that allow the development of scalable applications using a single code base, while also taking full advantage of the capabilities of the device. The market is large, but developers few!

The use of mobile computers is spreading faster than any other consumer technology in history. Smartphones outsell PCs. Touchscreens outnumber keyboards.

As per the Strategy Analytics Research Report on 'Global Mobile Workforce Forecast Update 2012-2018', worldwide, the mobile workforce is predicted to increase from 1.26 billion in 2013, accounting for 36.4 per cent of the global workforce, to 1.67 billion in 2018, accounting for 41 per cent of the global workforce.

Clearly, mobile apps are now being viewed by enterprises as core drivers that improve operational efficiency and have qualified as business-critical technology. However, app delivery is less about the technical aspects of programming and more about an app's business impact, design, integration, deployment and management.

This article will delve into the major challenges involved in designing, developing, deploying and maintaining mobile apps for enterprises, and how mobile application development platforms (MADP) solve some of these challenges.

Enterprise mobile apps: The challenges

Enterprise mobile apps involve some specific challenges that need to be addressed right in the planning and design phases. Later discoveries of one or more of these challenges have resulted in many enterprises having to re-plan and redesign their mobility solution roadmap. Let's take a deeper look at these challenges.

BYOD and device diversity

The Bring Your Own Device (BYOD) trend has penetrated

enterprises everywhere. If your enterprise mobile app is to support the BYOD strategy, then an altogether different level of challenges needs to be addressed. The target device spectrum grows exponentially when considering BYOD, with new devices entering the market and reaching your target users every day. Keeping track of mobile platforms, versions and new device models (on a variety of platform versions and differing configurations) makes supporting BYOD for your enterprise app a big challenge, without a well-planned approach.

Even if you do not opt for BYOD, the target device spectrum may be limited yet heterogeneous. Diverse mobile platforms (Android, iOS, etc), along with the dynamics they bring in, makes app development for supporting multiple platforms complex. Further, apart from the variety of devices that currently exist, more keep arriving in the market for each of the mobile platforms, thereby imposing the need for an enterprise to ensure that the mobile apps not only support required mobile platforms, but also work well on the target device spectrum per platform.

Enterprise data access

An enterprise mobile app may need to access data stored in a variety of heterogeneous data sources at the enterprise backend. For instance, the sales team's data may reside in the Salesforce CRM while the financial data resides in Oracle. Integrating with a variety of standard and custom data sources may be a big need when creating an enterprise mobile app.



It's Good To Have A Choice!

**ASSURED
GIFT WITH EACH
SUBSCRIPTION**

**GET
DISCOUNTS**

OR



Coffee Maker
(Worth Rs 1400)



EPFY Round Neck T-Shirt
(worth Rs 300)



OSFY Coloured T-Shirt
(Worth Rs 500)



Popcorn Maker
(Worth Rs 1100)



Toaster Oven
(Worth Rs 2000)

*Terms: 1. These rates are applicable for new subscribers as well as renewal by existing subscribers. 2. Gift items: prices are representative and actual product may vary. 3. The rates are valid for subscribers within India only. 4. Please allow 4-6 weeks for processing of your subscription. 5. Shipping charges are subject to local conditions of transport. Customs and duties in Delhi/New Delhi only. *Replacement will be made 7 (seven) days after receipt of copies of returned items. 30 days of its publication. **After three months, if you are not satisfied with the magazine, your balance amount will be returned.

SUBSCRIBE TO YOUR FAVOURITE MAGAZINES NOW!



Free e-zine Access With Every Subscription

Option 1: Assured Gifts with every subscription

Magazine	Duration	Issues	Cover Price (₹)	You Pay (₹)	Assured Gift
Electronics Bazaar	1 Year	12	1200	1200	☐ EPFY Round Neck T-Shirt (worth Rs 300)
	3 Years	36	3600	3600	☐ Popcorn Maker (Worth Rs 1100)
	5 Years	60	6000	6000	☐ Toaster Oven (Worth Rs 2000)
Open Source For You	1 Year	12	1200	1200	☐ OSFY Coloured T-Shirt (Worth Rs 500)
	3 Years	36	3600	3600	☐ Coffee Maker (Worth Rs 1400)
	5 Years	60	6000	6000	☐ Toaster Oven (Worth Rs 2000)

Option 2: Upto 40% Discount Offer

Magazine	Duration	Issues	Cover Price (₹)	Discount	You Save	You Pay (₹)
Electronics Bazaar	1 Year	12	1200	20%	240	960
	3 Years	36	3600	30%	1080	2515
	5 Years	60	6000	40%	2400	3595
Open Source For You	1 Year	12	1200	20%	240	960
	3 Years	36	3600	30%	1080	2520
	5 Years	60	6000	40%	2400	3600

Offer Valid till 31st October 2015

To Subscribe Online, Visit: <http://subscribe.etyindia.com> For Online Renewal, Visit: <http://renew.etyindia.com>

Name _____ Mailing Address _____

City _____ Pin Code _____ State _____

Email _____ Subscription No. (for existing subscribers only) _____

above (✓) marked magazine(s) starting with the next issue. Please find enclosed a sum of Rs _____ by DD/MO/crossed cheque/banking the

No. _____ dt. _____ in favour of EPFY Enterprises Pvt Ltd, payable at Delhi. (*Please add Rs 50 on non-metro cheque)

Send this filled-in form or its photocopy to:

EPFY Enterprises Pvt Ltd D-87/1, Disha Industrial Area, Phase 1, New Delhi 110 020

Ph: 011-26810881-82; Fax: 011-26817563; e-mail: info@ety.in

EPFY GROUP

www.ety.in



Introducing EPFY's Premium Edition
Electronics For You PLUS
(With software-packed DVD)
☐ 1 Year (12 issues) 1200 ₹ 960
☐ 2 Yrs (24 issues) 2400 ₹ 1800

Free e-zine Access With Every Subscription

- Don't Miss A Single Issue!
- Ensure Regular Supply
- FREE Replacement Policy*
- Order Now For FREE Home Delivery

*Replacement will be made 7 (seven) days after receipt of copies of returned items. 30 days of its publication. **After three months, if you are not satisfied with the magazine, your balance amount will be returned.

It would require considerable time and energy to enable the mobile app to directly access the variety of data sources being used at the enterprise backend.

Enterprise data security

With legacy systems and related security ecosystems, enterprise IT heads can remain relaxed because all the enterprise's data resides in and is mostly used from within the enterprise premises and via enterprise-owned devices like laptops, desktops, etc.

With mobility, data not only crosses beyond the enterprise premises, but often also resides on the users' personal mobile devices for offline access. Further, enterprise data communication happens mostly over untrusted networks on these mobile devices, which creates the risk of enterprise data being stolen or tampered with. Also, confidential enterprise data may also get copied, printed, etc. Other malicious apps may invade the enterprise app sandbox and access its data. The user may lose the device containing enterprise data, leading to a leak of the confidential data residing in the device. Developing a robust and reliable solution to secure both enterprise data at rest and data-on-the-move pertaining to the app (while not violating the user's privacy) is very crucial.

Remote management

Any loophole in an app's life cycle and its data management can make an enterprise vulnerable to mismanagement of app versions, users and their devices. For instance, the app should support a way to enforce app data wiping if a user is leaving the company or has lost the device. Further, if the enterprise wants to make a previous app version obsolete and prohibit its use, then it should be able to enforce specific app version updates as mandatory on the target user base.

Only complete control and management of the app's life cycle, together with the installed app and its data, can ensure the solution generates the desired return on investment (RoI) for an enterprise.

User adoption

In today's mobility era, an app's success depends, to a great extent, on its user experience. Streamlined use cases, ease of use, shorter workflows, a focus on only what the user needs in the current context, and a respect for screen size contribute to a better user experience. The better the user experience, the higher the chances of more users adopting the app. And the lower the user adoption, the higher the chance of an enterprise failing in its mobile app journey.

Enterprise mobility, as a segment, has been maturing over the past few years, and now there are solution suites available to address these challenges effectively. Enterprise mobility management (EMM) and the mobile application development platform (MADP) are well known solutions in the market. In this article, we will have a look at MADP and how it solves some of these challenges.

The solution: MADP

According to Gartner, "A MADP enables an enterprise to design, develop, test, deploy, distribute, analyse and manage a portfolio of cross-platform mobile apps on a range of devices running Android and iOS, and to address the requirements of diverse use cases including external-facing and internal-facing scenarios."

There are generic MADPs available in the market as products or by service providers as their solution catalysts. Whether to use a generic MADP or opt for a custom one largely depends on the complexity of your enterprise's ecosystem, its time-to-market needs, security concerns and customisation requirements.

Figure 1 shows the market leading MADPs, as per Gartner's Magic Quadrant for MADPs, July 2015.

As per this report, the MADP market continues to mature, moving away from the tech-centric position of the past towards a more business-aligned approach, expanding support for the app development life cycle and offering more robust and scalable capabilities.

As with all Magic Quadrants, vendors in the 'Leaders quadrant' are not necessarily the best for all projects or enterprises. Depending on an enterprise's needs, a vendor in any quadrant could be the best for an enterprise. Some platforms may be a better fit with an organisation's particular skill sets. Consider the match between the skills available and those required by each MADP evaluated. Also consider the cost and complexity of the vendor's offering, and whether this aligns with benefits expected during the estimated life span of the product.

Some enterprises may prefer custom MADP, to address specific concerns related to vendor lock-in, the nature of the solution, security, roadmap demands, long-term costs, etc. For developing custom MADP, enterprises can consider



Figure 1: Gartner Magic Quadrant for MADPs in 2015

leveraging a mix of freeware or open source components and custom components.

Recommendation: Before deciding on which open source tools and libraries to leverage, a precise check on their capabilities, roadmap, support services and costing, long term plans, customisation possibilities and licence type will help you in taking the most appropriate decision.

The MADP advantage

Enterprises leverage MADPs to get the benefits of one or more of the following salient features.

Excellent support for cross-platform mobile app development

This feature addresses the mobile platform diversity challenge by providing a way to write selected or common code/scripts for mobile app development targeted at all platforms. Internally, the platform may use a variety of approaches to convert the common code to platform-specific apps.

Many MADPs like PhoneGap provide a hybrid development framework, which provides the best of the Web (also known as thin client) and native (also known as thick client) approaches. It facilitates communication between the Web and the native parts of the app along with other rich features. This paradigm is characterised by the reuse of Web parts across mobile platforms while balancing the native approach benefits.

There are also MADPs like Appcelerator, which consist of inbuilt converters to convert the common script written by MADP developers into platform-specific native apps. In short, MADPs help to optimise the time, effort and cost of maintaining apps for multiple platforms.

Recommendation: MADPs differ in their support for specific mobile platforms and scripting languages, and come with their own limitations. An enterprise must analyse its existing or long-term preferred skill sets, vendor binding limitations and target mobile platforms to better judge which MADP suits it the best. It would also help to get a complete overview of the vendor MADP roadmap and match it with your enterprise's mobile app feature needs, in case you choose a standard MADP solution.

The standard MADP solutions roadmap and features are influenced by a variety of factors and your enterprise feature requirements may or may not match with them. This is why a large number of enterprises prefer customised MADPs to avoid vendor binding and to gain complete control as per their app roadmap, while keeping the solution simple and targeted to their specific needs.

Easy integration with standard enterprise data sources

This feature addresses the enterprise data access challenge to a large extent. Enterprise data sources may be standard

like SAP, Salesforce, JDEdwards, etc, or customised.

While MADPs provide loosely coupled and clean ways to integrate standard data sources, if custom data source integration is required, it would consume effort and time, depending on the complexity of integration.

Recommendation: Different MADPs feature integration with different sets of data sources. An enterprise looking for a MADP must map its specific data source integration needs with the data sources supported by MADPs, to take an appropriate decision.

Always think in terms of the future roadmap in order to cover the bigger picture of the mobility solution.

The power to manage the mobile app's lifecycle

This feature addresses the remote management challenge to some extent. While enterprises have a collection of apps planned or rolled out for different target user bases, rolling out and managing multiple apps for multiple platforms together with multiple versions of each becomes a huge challenge for the enterprise.

MADP solves this challenge by helping to design, develop, test, deploy, distribute, analyse and manage a portfolio of cross-platform mobile apps on a range of devices. In short, it brings all mobile app lifecycle related operations together in one solution for ease of management.

Recommendation: Different MADPs support different approaches, standards and tools for achieving the different lifecycle processes. The enterprise must match its existing tools and processes to the MADPs in consideration before taking the final call.

Easy integration with cloud services and third party tools

This feature addresses the dual challenges of enterprise data access and security. An enterprise may be leveraging certain cloud services and its data may reside on the cloud for a certain category of ecosystem components. Further, it may leverage third party tools of mobility management like Airwatch, MobileIron, etc. The enterprise mobile app must integrate well with these heterogeneous sources of data and the management tools.

Many MADPs consider this crucial need and provide good support for easy integration with these third party services.

Recommendation: The enterprise needs to understand its existing ecosystem and processes, together with its needs relating to third party tools integration with respect to MADP, before taking the final call. 

By: Pooja Maheshwari

The author works as an enterprise mobility evangelist with Transility (a division of Impetus). With 15 years of experience, she is a veteran in the enterprise mobility domain. Connect with Pooja at: pooja.a.maheshwari@iitbombay.org, <https://in.linkedin.com/in/poojamaheshwari99>

Why Are Organisations Moving to Microservices?

Microservices is a software architecture style in which complex applications are composed of small, independent, narrowly-focused processes communicating with each other using language independent APIs. Amazon and LinkedIn have both implemented this architecture, and others are following suit.

The word 'Microservices' emerged from a workshop on software architecture in May 2011, when, for the first time, architecture style was discussed. Very simply put, Microservices is like UNIX commands, which are designed to do one single thing but can be piped together to perform some complex operations. We can say that Microservices is loosely coupled, fine-grained, follows the Single Responsibility Principle [Robert C. Martin], is highly cohesive and autonomous. A more specific and clear definition by Martin Flower is, "The Microservices architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and are independently deployable by fully automated

deployment machinery. There is a bare minimum of centralised management for these services, which may be written in different programming languages and use different data storage technologies."

Service oriented architecture (SOA), continuous integration and deployment, domain driven development and automation have all led to the Microservices architecture. Netflix, Amazon, LinkedIn, Groupon, PayPal, Airbnb, The Guardian and many other companies have implemented or are moving towards the Microservices architecture.

To understand Microservices better, we need to understand the macro or monolithic application style first. According to Wikipedia, "A monolithic application describes a single-tiered software application in which the user interface and data access code are combined into a single program from a single platform." A monolithic application is built or packaged as a single



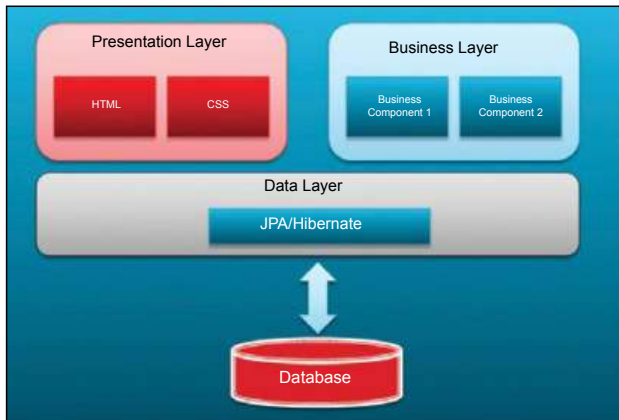


Figure 1: Three-layered monolithic application

executable or deployable unit. When deployed in a production environment, it runs as a single process. In Java terms, it is a WAR or EAR file, which is deployed on a Java EE container. These types of applications have modules, components and use one or more third party libraries. All such applications have three or more layers. How the layers commonly appear in a monolithic application can be seen in Figure 1.

The three layers are:

1. **Presentation layer:** This consists of a client side user interface mostly consisting of HTML, CSS, etc. It is also known as the UI layer.
2. **Business layer:** This consists of business components, REST, etc.
3. **Data layer:** This normally uses JPA, Hibernate, etc.

Figure 2 illustrates the advantages of the monolithic application.

Issues related to the monolithic application

This approach works fine, but when the application becomes large and the team grows in size the drawbacks or issues start surfacing. The common issues with this type of approach are summed up below.

Difficult to deploy and maintain

If, for instance, you have an application up and running as shown in Figure 1, and you change some business components as in Figure 3, you will need to rebuild your whole application and replace the old archive ear/war (WAR is for Web Archives and EAR is Enterprise Archive in Java) with the new one. Although you haven't changed anything in the other layers, you will have to redeploy the whole application. A small change needs to be rigorously tested to ensure it doesn't affect any other part of the application. So if anything changes in the monolithic application, there is no way we can test and redeploy that part only—the whole application has to be redeployed. If the application is huge, then the size of the ear/war file is quite large and will take time in deploying. And deploying the whole application again will interrupt

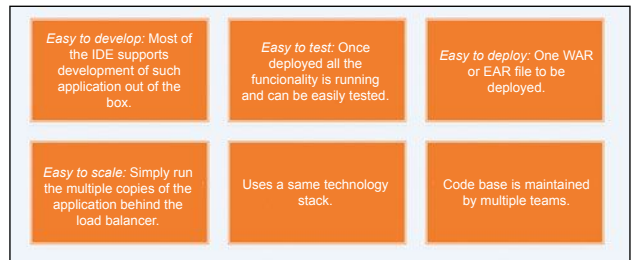


Figure 2: Advantages of the monolithic application

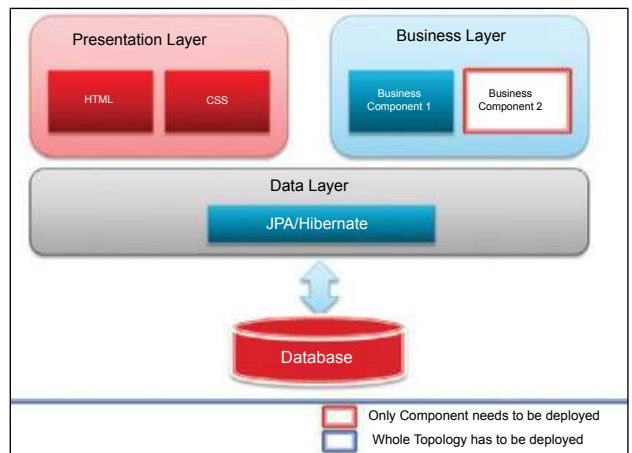


Figure 3: Changing the components in the business layer

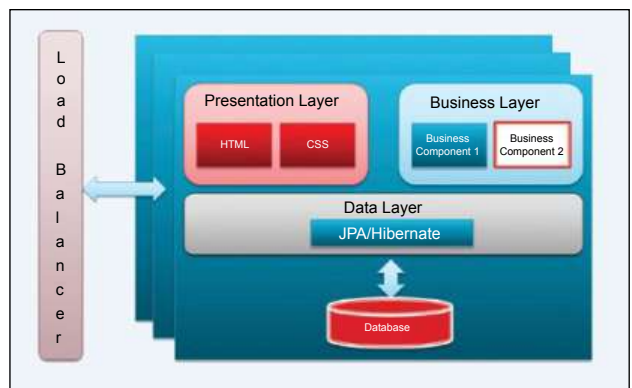


Figure 4: Scaling the application

the background tasks, irrespective of whether they are impacted by the change or not.

Stuck with the particular technology stack or framework

The other disadvantage is that we are stuck with the technology stack or framework for the long term. If any change is to be made in the framework or technology, the whole application will be affected and may need to be rewritten. For instance, if we have selected JVM, then we cannot think beyond it. Non-JVM languages and libraries cannot be used. Updating the JVM in the stack will lead to rigorous testing to ensure the application is working. If one component wants to use the latest feature

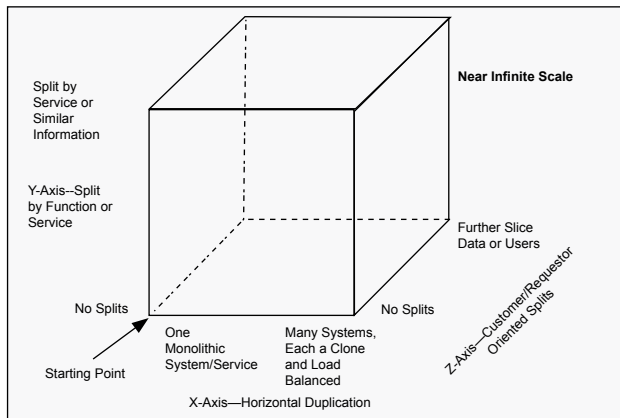


Figure 5: The Scale Cube (Source: The Art of Scalability by Martin L. Abbott and Michael T. Fisher)

from a third party library that supports Java 8, but other components are not compatible and are still dependent on Java 6, it's difficult to do so.

Scaling the whole application when only a certain part needs scaling

Scaling requires the entire application to be scaled rather than just parts of it. This requires greater resources as shown in Figure 4.

It is possible to scale horizontally by running many instances behind the load balancer. Scaling can even be done on the basis of sorting the request origin. For example, if the request comes from the Asia region, use a predefined DNS server or database server; if the request is from the European region, use the other DNS or database server, etc. The art of scalability shows scaling on three axes. When using the monolithic approach, it is not possible to scale on functional decomposition. Figure 5 shows three different axes for scaling.

Apart from this, when a new person joins a team, he or she needs to understand all the layers and has to go through thousands of lines of code so as to become a productive team member. Any new innovative idea may not be accommodated or be implemented easily, due to partitioning of the team on the basis of technology, which leads to a communication gap. Changes require a lot of approvals, which may take a long time.

The Microservices approach

The Microservices architectural style gained momentum because it allowed building the application as suites of services. Each service does exactly one thing or implements one functionality; it is independent of other services, can be deployed independently and can even use a different stack of technology – one that is the most efficient for implementing a particular functionality.

Characteristics of Microservices architecture

According to Martin Flower, the common characteristics of Microservices architecture are:

Componentisation via services: Instead of building components, we can think in terms of services, which can run independently and will expose themselves as REST end points or some other mechanism.

Organised around business capabilities: We normally make teams on the basis of technology. Almost all large organisations have a UI team, a database team, an engineering team, etc — hence the three-layer architecture [Figure 1]. Due to this, even a simple change needs to be communicated in a channel, and may take a long time for approval.

In the Microservices approach, the teams are cross-functional and are built according to business capabilities, such as the UI, database, project management, code, testing, etc. Each team is responsible for one service, which it builds and runs. The team is solely responsible for that service. The team size ideally follows the ‘two pizza rule’ that Amazon CEO Jeff Bezos came up with — the size of a team should not be larger than what two pizzas can feed.

Products not projects: In the monolithic approach, a project is built by a team, goes for testing and then moves to the maintenance phase. This is a typical project model.

In Microservices, the team is built around the service and the product it builds; the team owns it for its full life time. The Amazon concept is: “You build and you run it.”

Smart endpoints and dumb pipes: In the SOA based approach, there is the concept of an enterprise bus which is like the brain of the system. It does everything — from message routing and transformation to applying business rules, and so on.

In Microservices, on the other hand, the bus is dumb, as the services need to be decoupled and cohesive. It uses HTTP for communication, and may use REST or some other way to

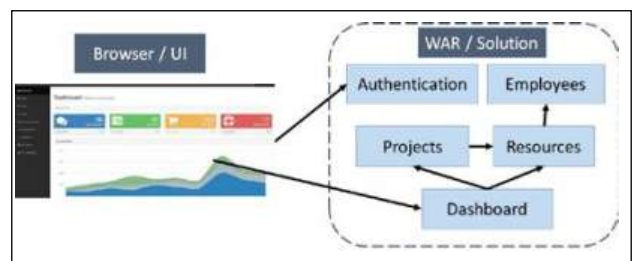


Figure 6: Conceptual monolithic application (Source: ‘Predicting the future of Microservices and SPA’ by Keyhole Software LLC)

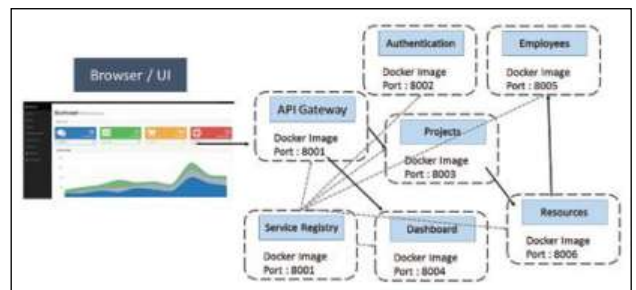


Figure 7: Monolithic application decomposed in Microservices (Source: ‘Predicting the future of Microservices and SPA’ by Keyhole Software LLC)

achieve the same results.

Decentralised governance: The monolithic application standardises on a single technology platform which may seem healthy to start with but, later, it may become a bottleneck when some of the modules or components need to use other styles.

In Microservices, each team can select a different standard for the service it is building. Netflix is a good example.

Decentralised data management: In Microservices, instead of using the same database, each service can decide its own storage. For example, LinkedIn was using Oracle when it was running as a monolithic application, but then moved to Microservices and started using the graph database for storing people and their connections, apart from Oracle. Microservices prefers that each service manages its own database.

Designed to cope with failure: When we think of service as a component, the application on the whole is to be designed to cope with the failure of the service. As this is mostly going to be a network call using an end point which is bound to fail, it should be reflected upon. Netflix uses SimianArmy to test failure.

Evolutionary design: When we think of Microservices, we think in terms of decomposing the system into small services. The key part is the notion of independent replacement and upgradeability. The Guardian website is moving towards Microservices, due to which it can

easily add quick new sections for new events which can be removed quickly once the event is over, as it is a service.

To get a clearer picture, look at Figure 6, which shows a conceptual monolithic application and Figure 7, which depicts the application split into Microservices.

The Microservices approach has some drawbacks like dealing with the complexity of a distributed system. The two common points of failure are services failing to respond and high latency. Transaction management in distributed systems is difficult to maintain. It brings lot of operational overhead, and the team should be ready to manage and monitor a lot of services which are redeployed continuously. If the organisation does not have the Dev-Ops culture, moving towards Microservices is bound to fail. 

References

- [1] <http://www.martinfowler.com/articles/microservices.html>
- [2] <http://microservices.io/>

By: Ashish Singh Bhatia

The author is a technology enthusiast and a FOSS fan. He loves to explore new technology, and work on Python and Java languages. He can be reached at ast.bhatia@gmail.com and blogs at <https://openfreeidea.wordpress.com/>

Continued from page 77...

```
$ git cherry-pick -m master <commit-id>
/* Test again in release version to make sure */
/* Push to official-repository */
```

Note that since we had squashed multiple commits into a single logical commit for the defect concerned, as part of our development workflow, only a single commit needed to be cherry-picked. If *squash* was not done, we would have had to track and cherry-pick multiple commits.

Common errors while using Git and how to recover from them

There are times when we make mistakes, but the power of Git helps us to recover from them. However, for the following actions, do read up the documentation and understand what they do and what the side-effects could be, before use.

- To undo a wrong change in a file which is not committed, use the command given below:

```
$ git checkout -- <file>
```

- If you missed committing a file from a commit that you just made, issue the following command:

```
$ git add <missed-file>
$ git commit -amend
```

- If you wrongly committed some change, use the command shown below:

```
$ git reset HEAD-1
```

- If you messed up and want to go to a clean slate without having to clone again (i.e., you don't want generated or untracked files around), use the following code:

```
$ git clean -xfd
```



References

- [1] *Git Magic*: <http://www-cs-students.stanford.edu/~blynn/gitmagic/>
- [2] *Pro Git*: <http://git-scm.com/book>
- [3] *Online reference*: <http://git-scm.com/docs>
Book: 'Version Control with Git', Jon Loeliger and Matthew McCullough, O'Reilly Publishers.

By: Natarajan Venkataraman

The author is an industry veteran with over two decades of professional experience in embedded systems and networking domains. His areas of interest include quality of service, security, algorithms and last, but not the least, good software development practices. He can be reached at v_natarajan@acm.org.

OpenModelica: A Powerful Engineering Modelling and Simulation Tool

The use of modelling and simulation in engineering has long been an accepted practice. Models represent the real systems either as physical reproductions in a smaller scale, or as mathematical and computer models that allow representing the dynamics of the system via simulation. Models enable you to study a system's behaviour in ways that are either not possible or too risky in the real world.

Modelling is at the heart of modern engineering practice. Engineers model things all the time. Models allow engineers to conduct many 'what if' experiments and also to study potential failures in the system. While mathematical models have been in use for a long time, in the last few decades, computer models have become indispensable. But one problem remains — the different ways in which engineers from different streams build the models. For example, mechanical engineers usually use CAD models and Finite Element Analysis. Electrical engineers use circuit diagrams, while a computer science engineer may use flow charts and object-oriented modelling in programming.

There have been attempts to bridge this gap, though. One particular solution is the Modelica language, which is for engineering design and simulation. Modelica is a domain neutral language with which small, simple things like resistors, as well as large and complex systems like automobiles and power plants can be modelled.

The Modelica language

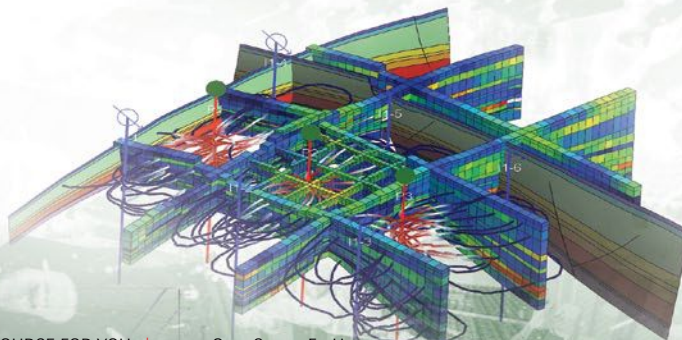
Modelica is an object-oriented, equation based modelling language. It is not proprietary, and is managed by the Modelica Association. The purpose of Modelica is to describe engineering components and systems. We cannot think of Modelica as a programming language such as C++ or Java. Modelica needs an implementation environment. It is important to remember this — Modelica is a modelling language. Once we have modelled an engineering system, we can run the model and see how the system behaves. For example, if you've built a model of a braking system for a car, you can run the simulation and see how the braking system behaves under different load conditions. Or if you have built a circuit model, when you simulate the model, you can see how current and voltage change with respect to time

in different components. Modelica model files have .mo as the extension.

As mentioned earlier, we'll need to implement an environment for Modelica models to run. There are many such run environments. Some are open source, and some are proprietary. Important run environments for Modelica are OpenModelica, JModelica and Dymola. While OpenModelica and JModelica are free and open source platforms, Dymola is a commercial platform. Personally I would prefer OpenModelica as it is cross-platform, easy to use/install, comes with many other supporting tools, has very good documentation for learning, and has a reasonably good community around it, which is very essential.

Modelica's components library

Modelica's components library is a very powerful feature. As Modelica is an object-oriented modelling language, the idea of re-using components is very naturally exploited. And to make things even more interesting, Modelica Association releases what's in the components library on a free basis. The library is basically a collection of models of the most commonly used engineering components. To be more specific, it contains Blocks (mathematical/logical function blocks), ComplexBlocks, StateGraph, Electrical, Magnetic, Mechanics, Fluid, Media, Thermal, Math, ComplexMath, Utilities, Constants, Icons, and SIunits packages. Each of these 'packages' contain different component models under different sub-sections. Now imagine that you're building a model of a robotic arm, which contains both electrical and mechanical components, along with control software. You don't have to build models of components like the motor or shaft. You can just use the available component models in the library, fine tune the model parameters and 'connect' to see how your system works.



OpenModelica

OpenModelica is a platform for the Modelica language. It's free, based on open source software and is supported by the Open Source Modelica Consortium. OpenModelica can be installed both on Linux and Windows platforms. In the OpenModelica package, three important modules are available – OMEdit, OMNotebook and OMShell. OMEdit is a graphical interface, which can be used to build models and run simulations. OMNotebook is an innovative learning companion, with which you can read the relevant information on a particular concept and also run the code there itself. While it will not teach you how to use OMEdit, which is a graphical editor, OMNotebook will help you to understand the basics of the Modelica language very clearly and help you build a very strong foundation on the topic. You can download and install OpenModelica for Linux from <https://openmodelica.org/download/download-linux>, and for Windows from <https://openmodelica.org/download/download-windows>.

If you face a problem in installing it, you can also download a virtual machine with pre-installed OpenModelica from <https://openmodelica.org/download/virtual-machine>.

Getting started with OMEdit

OMEdit is a graphical editor that allows you to access Modelica's components library, which you can see on the left side of the screenshot in Figure 1. Building a model is as easy as dragging and dropping components into the 'model' area, and connecting suitable components.

Of course, you should know which components to use, and what the appropriate value for different parameters of the components is. We'll build a very simple electric circuit to get started with OMEdit. Here are the steps.

1. Start OMEdit.
2. Go to *File* → *New Modelica Class* → *Name* → *ResistorCircuit* (we've provided model name, not file name).
3. Click on *Save* → *ResistorCircuit.mo* [Note: This is the file name].
4. Go to Modelica's components library. Open *Electrical* → *Analog* → *Basic*. Drag and drop resistors into the model area. Double-click, and enter resistor values for both.
5. Similarly, open *Electrical* → *Analog* → *Sources* → *Constant Voltage* (for batteries).
6. Once the components are in place, and the appropriate parameter values are entered, we're ready to connect. To do so, click, drag, and connect from one end of the component to another end.
7. Once all the connections are done, run the simulation.

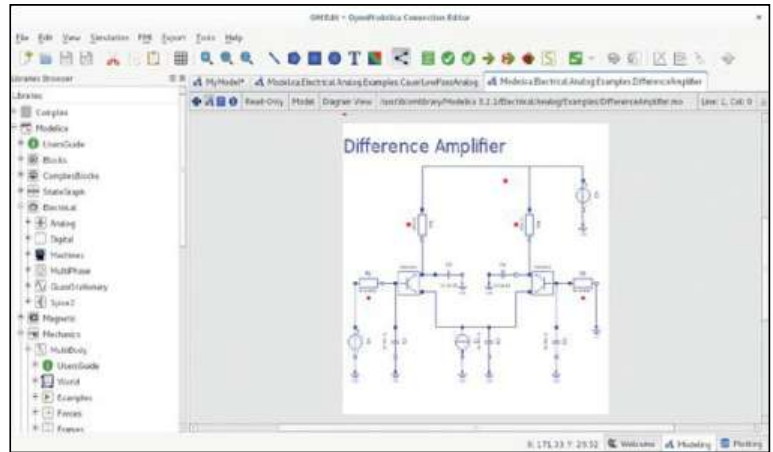


Figure 1: OpenModelica

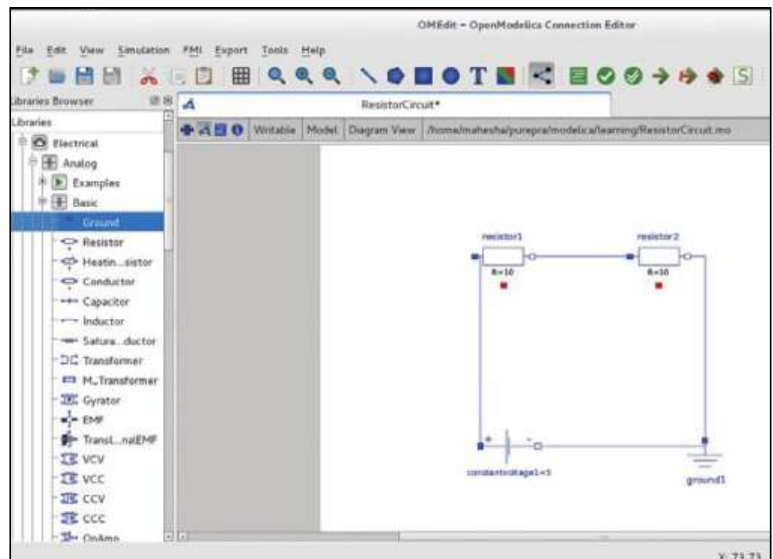


Figure 2: Resistor circuit

Menu: *Simulation* → *simulate*

8. Viewing the results is also very easy. Just click on the *Plotting* tab in the bottom right corner, and you'll see a list of components in a tree. If you expand the tree, you can select the parameter of your choice, and you'll be able to see the plotting of that parameter value with respect to time. While this example used only electrical components, we can easily use mechanical and electrical components in a single model and simulate the operations. I would also recommend that users explore OMNotebook, as it can be a very good learning companion in the beginning. **END** 

By: Mahesha Hiremath

The author is an independent data analytics consultant from Bengaluru. He is building a platform for electric vehicles-Project Vidyut. Project Vidyut is a not-for-profit initiative for promoting electric vehicles, and also to build open source electric vehicles. He can be reached at mahesha-at-projectvidyut-dot-org and tweets with [handle@MaheshaHiremath](https://twitter.com/MaheshaHiremath).



Having explored ns-3 for the past five months in *OSFY*, we now have come to the concluding article in the series. Through the series, awareness of ns-3, installation, its various tools and, finally, the traps and pitfalls associated with it have been covered. No study can ever encompass the full range of ns-3's potential. We hope some of our readers will now be motivated enough to seek more knowledge about ns-3 and apply it.

This is the last article in the series on ns-3 and I would like to use Winston Churchill's quote from an entirely different context: "Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning." He was referring to the second battle of El Alamein, Egypt, when the Allied army defeated the Axis powers. Yes, the series is ending but our endeavour to master ns-3 continues. Over the past issues, I tried to give you an overview of ns-3. This tutorial is not complete, but the material I have covered is sufficient to initiate you into the intricacies of ns-3. Before we wind up, let's try to tie up a few more loose ends.

Python in ns-3

I have mentioned that the use of Python is not mandatory in ns-3. But I need to explain the significance of Python in ns-3, because those who are familiar with Tcl in ns-2 expect a similar role for Python in ns-3. This is a misconception. The ns-3 and ns-2 simulators are independent of each other and

the way Python interacts with ns-3 is entirely different from the way Tcl interacts with ns-2.

So what is the relationship between Python and ns-3? Are they mere acquaintances, friends or brothers? A technique called Python bindings is used in ns-3 to write ns-3 simulation scripts in Python. Python binding is a general technique used to access libraries in other programming languages from Python. For example, it is possible to create a binding to access C++ libraries from Python. But we already know how to simulate ns-3 scripts directly using C++. Then why is a second language like Python used to replicate the same results? Well, right now, that's all Python is offering. The interaction between Python and ns-3 is still a work in progress. But when Python bindings in ns-3 are complete, as predicted, we will have the ability to create new prototype models in ns-3, using Python alone. For example, we might have the ability to simulate a new routing protocol using Python without modifying the C++ source files of ns-3. But until then, the use of Python in ns-3 is not highly recommended for beginners.

Anyway, let's discuss how Python bindings help us execute ns-3 (the source code of which is in C++) with Python code. In this aspect, the use of Tcl in ns-2 and Python in ns-3 have some similarity. But the main difference is that the execution of an ns-2 simulation is possible only with Tcl scripts, whereas the execution of an ns-3 simulation is possible even without using Python scripting. The Python script shown below called *pythonscript.py* executes C++ based ns-3 simulation. This is a slightly modified version of a Python script available in the standard set of examples provided with ns-3. You can download the file *pythonscript.py* and all the other files mentioned in this article, from http://opensourceforu.ifytimes.com/article_source_code/October2015/ns3.zip.

```
import ns.applications
import ns.core
import ns.internet
import ns.network
import ns.point_to_point
ns.core.LogComponentEnable("UdpEchoClientApplication",
ns.core.LOG_LEVEL_ALL)
ns.core.LogComponentEnable("UdpEchoServerApplication",
ns.core.LOG_LEVEL_ALL)
nodes = ns.network.NodeContainer( )
nodes.Create(2)
pointToPoint = ns.point_to_point.PointToPointHelper()
pointToPoint.SetDeviceAttribute("DataRate", ns.core.
StringValue("10Mbps"))
pointToPoint.SetChannelAttribute("Delay", ns.core.
StringValue("1ms"))
devices = pointToPoint.Install(nodes)
stack = ns.internet.InternetStackHelper( )
stack.Install(nodes)
address = ns.internet.Ipv4AddressHelper( )
address.SetBase(ns.network.Ipv4Address("192.168.100.0"),
ns.network.
Ipv4Mask("255.255.255.0"))
interfaces = address.Assign(devices)
echoServer = ns.applications.UdpEchoServerHelper(5000)
serverApps = echoServer.Install(nodes.Get(1))
serverApps.Start(ns.core.Seconds(2.0))
serverApps.Stop(ns.core.Seconds(8.0))
echoClient = ns.applications.UdpEchoClientHelper(interfaces.
GetAddress(1), 5000)
echoClient.SetAttribute("MaxPackets", ns.core.
UIntegerValue(1))
echoClient.SetAttribute("Interval", ns.core.TimeValue(ns.
core.Seconds(1.0)))
echoClient.SetAttribute("PacketSize", ns.core.
UIntegerValue(512))
clientApps = echoClient.Install(nodes.Get(0))
clientApps.Start(ns.core.Seconds(3.0))
clientApps.Stop(ns.core.Seconds(8.0))
ns.core.Simulator.Run( )
```

```
ns.core.Simulator.Destroy( )
```

The Python script *pythonscript.py* simulates a network similar to the one we have simulated with the ns-3 script *simulation1.cc*, which has two nodes connected by a point-to-point link, and a UDP echo server and client as applications. Please refer to Part 3 (*OSFY*, July 2015) of this series to refresh your knowledge about the simulation script *simulation1.cc*. There are two different types of Python bindings in ns-3. These are the monolithic bindings that provide API definitions for all the modules and can be found in the directory *ns/ns-allinone-3.22/ns-3.22/bindings/python*, and the modular bindings which provide API definitions for individual modules and can be found in the respective modules themselves.

The ns-3 APIs are included into a namespace by Python bindings. Earlier, Python bindings used to generate API definitions for ns-3 modules into a namespace called 'ns3'. Later on, Python bindings started generating API definitions for ns-3 modules into a namespace called 'ns'. You can observe that in the script *pythonscript.py*, the namespace called 'ns' is used. The rest of the Python script is similar to the ns-3 script *simulation1.cc*. The Python script creates two nodes connected by a point-to-point link. Then the data rate and delay associated with the link is set. IPv4 addresses are assigned to the two nodes with the following line of code:

```
address.SetBase(ns.network.Ipv4Address("192.168.100.0"),
ns.network.
Ipv4Mask("255.255.255.0"))
```

This convoluted statement to assign IPv4 addresses to nodes is necessary due to another drawback of Python bindings. Some of the conversion constructors in the C++ source of ns-3 are yet to be supported by Python bindings. Thus, we have to use explicit constructors like the one shown in the previous line of code. After assigning an IP address on both the nodes, a port number is assigned on the node containing the UDP echo server application. The port number assigned to the server application is 5000. After this, a UDP echo client is set up on the second node. UDP echo client parameters like the maximum number of packets sent, the packet size, packet sending intervals, etc, are defined next. The starting and stopping times of both the server and client applications are also set. The lines of code '*ns.core.Simulator.Run()*' and '*ns.core.Simulator.Destroy()*' start and stop an instance of the ns-3 simulator.

Copy the file *pythonscript.py* to your system for execution. I have copied the file into the directory *ns/ns-allinone-3.22/ns-3.22/scratch*. Now open a terminal in the directory *ns/ns-allinone-3.22/ns-3.22* and execute the following command:

```

UdpEchoServerApplication:UdpEchoServer(0x8a09cb0)
UdpEchoClientApplication:UdpEchoClient(0x8a09ff8)
UdpEchoClientApplication:SetDataSize(0x8a09ff8, 512)
UdpEchoServerApplication:StartApplication(0x8a09cb0)
UdpEchoClientApplication:StartApplication(0x8a09ff8)
UdpEchoClientApplication:ScheduleTransmit(0x8a09ff8, +0.0ns)
UdpEchoClientApplication:Send(0x8a09ff8)
At time 3s client sent 512 bytes to 192.168.100.2 port 5000
UdpEchoServerApplication:HandleRead(0x8a09cb0, 0x8a0a2a8)
At time 3.00143s server received 512 bytes from 192.168.100.1 port 49153
Echoing packet
At time 3.00143s server sent 512 bytes to 192.168.100.1 port 49153
UdpEchoClientApplication:HandleRead(0x8a09ff8, 0x8a0a468)
At time 3.00287s client received 512 bytes from 192.168.100.2 port 5000
UdpEchoClientApplication:StopApplication(0x8a09ff8)
UdpEchoServerApplication:StopApplication(0x8a09cb0)
UdpEchoClientApplication:DoDispose(0x8a09ff8)
UdpEchoServerApplication:DoDispose(0x8a09cb0)
UdpEchoServerApplication:~UdpEchoServer(0x8a09cb0)
UdpEchoClientApplication:~UdpEchoClient(0x8a09ff8)

```

Figure 1: Output of the Python script

```

[root@deepu ns-3.22]# tcpdump -nn -tt -r pcapsim-0-0.pcap
reading from file pcapsim-0-0.pcap, link-type PPP (PPP)
3.000000 IP 192.168.100.1.49153 > 192.168.100.2.2000: UDP, length 512
3.010867 IP 192.168.100.2.2000 > 192.168.100.1.49153: UDP, length 512
6.000000 IP 192.168.100.1.49153 > 192.168.100.2.2000: UDP, length 512
6.010867 IP 192.168.100.2.2000 > 192.168.100.1.49153: UDP, length 512
[root@deepu ns-3.22]# tcpdump -nn -tt -r pcapsim-1-0.pcap
reading from file pcapsim-1-0.pcap, link-type PPP (PPP)
3.005433 IP 192.168.100.1.49153 > 192.168.100.2.2000: UDP, length 512
3.005433 IP 192.168.100.2.2000 > 192.168.100.1.49153: UDP, length 512
6.005433 IP 192.168.100.1.49153 > 192.168.100.2.2000: UDP, length 512
6.005433 IP 192.168.100.2.2000 > 192.168.100.1.49153: UDP, length 512

```

Figure 2: Output of tcpdump

```
./waf --pyrun scratch/pythonscript.py
```

Please note that the directory *scratch* is not the default location to save Python scripts and hence it is not mandatory to save Python files in this directory. Also remember to use the path starting from the directory *ns/ns-allinone-3.22/ns-3.22* to the directory containing the Python script. Since I have copied the Python script into the directory *scratch*, the path in the command is *scratch/pythonscript.py*. We have used the option '*LOG_LEVEL_ALL*' in the simulation script. Thus all the log information is displayed on the terminal. Figure 1 shows the terminal displaying the log information from the Python script.

The Python bindings feature is a work in progress and there are some problems associated with it. One is the insufficient coverage of ns-3 C++ source files. There are many ns-3 APIs yet to be supported by Python bindings. Newly created modules in ns-3 will not be supported by Python bindings. A tool called *pybindgen* is used to generate Python bindings for ns-3 source code written in C++. If you feel that all this fuss about the use of Python in ns-3 is unnecessary, then you can simply disable Python in ns-3. Open a terminal in the directory *ns/ns-allinone-3.22/ns-3.22* and execute the following command:

```
./waf configure --disable-python
```

On execution of this command, Python will be disabled in ns-3, and you can continue executing ns-3 simulation with the help of C++ simulation scripts as discussed in the previous parts of this series.

Tracing with pcap

We have earlier discussed ASCII based tracing in ns-3. But the ASCII trace file format is not the only one available in ns-3. It is possible to obtain a pcap trace file from an ns-3 simulation. An application programming interface for capturing network traffic is called pcap. A clear advantage of pcap tracing over ASCII tracing is that pcap trace files need not be processed manually. A lot of information can be obtained from pcap trace files with the help of utilities like tcpdump, Wireshark, etc.

If pcap tracing is such a treasure trove of information, why did I hide it till the end? Well, for the simple reason that pcap trace files are processed by third party utilities. It took me a long time to start discussing ns-3 in this series. So then, I thought it was not the right time to discuss pcap tracing and introduce yet another utility. But since we are about to wind up this series, I now believe the time is ripe to learn pcap tracing. In Linux, pcap is implemented in the *libpcap* library. To discuss the features of pcap tracing I will again use the simulation script *simulation1.cc* with which we are quite familiar. I have renamed the simulation script as *pcapsim.cc*. The only significant change made in the script *pcapsim.cc* is the addition of the following line of code:

```
pointToPoint.EnablePcapAll("pcapsim");
```

It enables pcap tracing in the ns-3 script. The extension of pcap trace files is *.pcap*. But the name of the pcap trace file generated is not *pcapsim.pcap* because naming conventions for pcap trace files and ASCII trace files are different. Unlike ASCII tracing, multiple trace files are generated in pcap tracing depending on the number of nodes in the simulation and the number of devices installed on these nodes. For example, if the simulation contains four nodes with each node having three devices installed on them, then a total of 12 pcap trace files will be generated. Thus, the number of pcap trace files generated equals the total number of devices in the simulation. In pcap tracing, the name given inside the function *EnablePcapAll()* in double quotes is just a suffix for the different pcap trace files generated. The name of each pcap trace file will be as follows: *file_name_suffix-node_number-device_number.pcap*.

In the simulation script *pcapsim.cc* there are only two nodes, and each has just one device installed on it. So, on execution, the simulation script will generate two pcap trace files named *pcapsim-0-0.pcap* and *pcapsim-1-0.pcap*. The execution of the simulation script *pcapsim.cc* is no different from the execution of all the other C++ simulation scripts in ns-3. Copy the file *pcapsim.cc* in the directory *scratch* and open a terminal in the directory *ns/ns-allinone-3.22/ns-3.22*. Execute the commands '*./waf*' and '*./waf --run pcapsim*' on the terminal. This will generate the two pcap trace files mentioned above. By default, the

pcap trace files will be saved in the directory *ns/ns-allinone-3.22/ns-3.22*.

The pcap trace files will not give you any information if opened in a text editor. To elicit information, you have to process these files with

a packet-analysing utility like tcpdump or Wireshark. First, let us see how to use tcpdump to extract information from the pcap trace files. The utility tcpdump is a packet analyser that runs in the command line. It has the ability to generate pcap trace files as well as analyse it. Since pcap trace files are already generated by ns-3, tcpdump is used only for analysis. Open a terminal in the directory *ns/ns-allinone-3.22/ns-3.22* and execute the following command:

```
tcpdump -nn -tt -r pcapsim-0-0.pcap
```

This command will process the pcap trace file *pcapsim-0-0.pcap*. Figure 2 shows the output of applying tcpdump on the two pcap trace files *pcapsim-0-0.pcap* and *pcapsim-1-0.pcap*.

The option *-tt* instructs tcpdump to print unformatted time on each line of the tcpdump output. The option *-r* instructs tcpdump to read and process an input file. As mentioned earlier, tcpdump has the ability to observe a network and produce a pcap data file. But here, pcap trace files are already available and hence the option *-r* is used. The option *-nn* instructs tcpdump not to convert the IP address to the hostname or the port number to the port name. For example, the IP address *64.233.166.94* will be converted to *google.co.in* and port number *80* will be converted to *HTTP* (remember, port number *80* is the well known port for HTTP) if the option *-nn* is not used in tcpdump. If you use the option *-n* instead of *-nn*, then the port number will be converted to the corresponding port name. Assume that UDP echo server port number was *80* in the script *pcapsim.cc*, then the command '*tcpdump -n -r -tt pcapsim-1-0.pcap*' will give the following output on the terminal:

```
3.005433 IP 192.168.100.1.49153 > 192.168.100.2.http: UDP,
length 512
```

```
3.005433 IP 192.168.100.2.http > 192.168.100.1.49153: UDP,
length 512
```

```
6.005433 IP 192.168.100.1.49153 > 192.168.100.2.http: UDP,
length 512
```

```
6.005433 IP 192.168.100.2.http > 192.168.100.1.49153: UDP,
length 512
```

As you can see, port number *80* is replaced by the port

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.100.1	192.168.100.2	UDP	512	source port: 49153 destination port: cisco-sctp
2	0.010867	192.168.100.2	192.168.100.1	UDP	512	source port: cisco-sctp destination port: 49153
3	1.000000	192.168.100.1	192.168.100.2	UDP	512	source port: 49153 destination port: cisco-sctp
4	3.010867	192.168.100.2	192.168.100.1	UDP	512	source port: cisco-sctp destination port: 49153

Figure 3: Wireshark window showing pcap trace file

name *http*. Since tcpdump is a very powerful utility, a lot of information can be extracted from pcap trace files by using the different options of tcpdump. But many new users of Linux and ns-3 feel a general aversion towards tcpdump because it is a command line utility. If you, too, feel the same way about command line utilities then Wireshark is the solution to your problems.

Wireshark is a packet analysis utility with a graphical front-end (GUI interface). You can open the pcap trace files in Wireshark and view the results on a window. Figure 3 shows the Wireshark window showing the pcap trace file *pcapsim-0-0.pcap*. In the figure, you can observe the fact that registered port number *2000* is identified with the port name *cisco-sctp* by Wireshark. A lot of options for filtering and analysis are available with Wireshark, which can be used to extract lots and lots of useful information from pcap trace files. A detailed discussion about Wireshark is beyond the scope of this tutorial but you can find a number of Wireshark related documents at www.wireshark.org/docs/.

A few words about the ns-3 source code

We have used many built-in classes of ns-3 to simulate different networks. What about the source code of these built-in classes? We have already discussed Doxygen and how it helps to maintain the documentation of ns-3. You can find the ns-3 documentation at www.nsnam.org/doxygen/. You can get details regarding ns-3 modules, namespaces, classes, etc, from there.

But what is the importance of ns-3 source code? We know for a fact that many network protocols and network related concepts can be simulated in ns-3. But what if your research project involves the creation of a new network protocol? Then you need to alter the source code of ns-3 to incorporate your new protocol into the existing structure of ns-3. So it is absolutely essential to go through the source code of ns-3. But remember, there are hundreds of thousands of lines of code in ns-3 and it is not humanly possible to go through all of it alone. So you must identify the ns-3 source files relevant to your research, have a deeper understanding of the code in them, and finally modify them. Once you have made changes to the ns-3 source code, you should rebuild the entire source code so that the modified code will have its effect on future simulations. If you have successfully altered the source code of ns-3 to create a new protocol, please share it with

Stay updated
with the latest
products in:

Surface Mount
Technology

LED

Electronics
Components

Power
Electronics

Test &
Measurement

Security &
Surveillance

Solar

Read
Electronics Bazaar

or Visit
www.electronicseb2b.com

the others so that they, too, can have a better solution to their research problem—just like the efforts of countless individuals who developed ns-3 helped us.

I would now like to mention a few topics that didn't make it into this series of articles on ns-3. The one that comes to my mind first is PyViz, a live simulation visualiser. It doesn't require any trace files to generate the animation. PyViz is mostly written in Python, but works with both Python scripts and C++ scripts. It is mostly used for debugging purposes. With it, we can check the working of mobility models and the time at which packets are dropped. PyViz will not work without Python bindings. So we have one point in favour of Python bindings.

Another area I skipped altogether is the installation of ns-3 using Cygwin (a utility which helps Linux applications to run on Windows). Over the years, I have faced the following question many times. Is it possible to install ns-2 and ns-3 on Microsoft Windows operating systems? Yes. It is possible to do so. But the question is, why? Cygwin based ns-3 installation is a tedious task. There might be compatibility issues between POSIX APIs and Win32 system calls. You might come across unexpected errors frequently. People hope that by installing ns-3 on Windows, the whole process of network simulation will become easier. This is a false hope; whether the installation is on Linux or Windows makes no difference to ns-3.

Another area I haven't covered is third party contributions to ns-3. Many research groups have contributed to the code base of ns-3. For example, cognitive radio extension for ns-3 called CRE-NS3 is maintained as contributed code by third party developers. There might be other interesting topics from ns-3 left out in this series for want of space. Out there is an ocean and I have just provided a cupful, so do forgive me.

You can find many useful ns-3 documents at www.nsnam.org/documentation/. Before concluding, a word about the way this series was organised. I haven't explained the finer details of ns-3; for that you have a tutorial available in the above mentioned location. I haven't elaborated much about the complex structure of ns-3; for that you have the manual available in the same location. My intention was to present the case for ns-3, guide every one through the installation process, familiarise every one with the different tools associated with ns-3 and, finally, warn you about a few traps and pitfalls in ns-3—most of which I myself have fallen prey to. **END!** 

By: Deepu Benson

The author currently works as an assistant professor in Amal Jyothi College of Engineering, Kanjirappally, Kerala and has 15 years of programming experience. You can contact him at deepumb@hotmail.com or at his technical blog www.computingforbeginners.blogspot.in.

Using GNU Emacs as an Internet Relay Chat

Internet Relay Chat (IRC) provides instant communication over the Web. It is generally used for one-to-one communication or group discussion forums, also known as channels. In this article in the GNU Emacs series, we will learn how to use GNU Emacs as an Internet Relay Chat (IRC) client.



Internet Relay Chat (IRC) is a messaging protocol that was created by Jarkko Oikarinen in 1988. An IRC server has a number of channels (or rooms, as they are called in other chat software) where both technical and non-technical discussions take place. Every user requires a nickname to chat. You will need to register your nickname with a password to identify yourself every time you log in to the IRC server. IRC can also be used for file sharing among users.

You can connect to a number of freely, available IRC servers. The most popular is Freenode (irc.freenode.net). The other IRC servers, to name a few, are IRCNet, OFTC, and EFNet. The Debian and Ubuntu projects have their own chat servers — irc.debian.org and irc.ubuntu.com. All IRC channel names begin with '#'. Some channels begin with '##'. Examples of channels are ##linux-india, #dgplug, #emacs, #ruby, #guile, #lisp, and #scheme. You can also create your own channels, or host your own IRC server. Some examples of free and open source IRC server software are IRCd, UnrealIRCd and Synchronet. Each IRC channel has a topic that can include useful links relevant to the channel. Sometimes, announcements or news updates are also mentioned in the topic.

Basic commands

A number of commands can be given to the IRC server. A few of them are discussed below:

1. `/list` is used to provide a list of all the available channels in a server. For example, `/list` in irc.freenode.net returned the following:

```
...
#linod      1
##welding   3      Welcome to ##welding, We're a little
bare at the moment, but will help if we can. Tutorials:
https://www.youtube.com/channel/UCJAFY2kKKb5sg79yld7T3hA
#drupal-ph  1      "Drupalista! Welcome to Philippine
Drupal Users Group. Have a good time chatting. If you have a
question, please don't ask to ask but fire up your question
in very specific and constructive way! Please join #drupal or
#drupal-support if no one is around"
#orx-project 4      Orx: Portable Game Engine
#tinkerforge 5
#osi        10     The Open Source Initiative
#xampp      1
#guitar     8
#bitcoin-ar  3      Comunidad Bitcoin Argentina
#LargeHadrosaurCollider 19  Welcome to the LHC, est. 2001
| http://www.largehadrosaurcollider.net | August Birthdays:
Digby 08/21, Josh 08/31 | At night it is pitch black, often
for months. | http://tinyurl.com/psgdagl
* End of /LIST
```

2. `/msg NickServ REGISTER password e-mail` is used to register your nickname to the IRC server. `/msg NickServ IDENTIFY password` is used to identify yourself to the server.
3. `/me message` displays the message for a user. For example:

```
/me says "Hello, World!"
```

* mbuf says "Hello, World!"

4. `/whois nickname` provides useful information for a user.
For example:

```
/whois mbuf
```

```
* [mbuf] (~shakthi@123.123.123.123): Shakthi Kannan
* [mbuf] #guile #scheme ##linux-india #stumpwm #guix #dgpplug
#lisp #emacs
* [mbuf] kornbluth.freenode.net :Frankfurt, Germany
* [mbuf] is connecting from *@123.123.123.123 123.123.123.123
* [mbuf] idle 00:41:52, signon: Thu Sep 3 20:36:52
* [mbuf] is logged in as mbuf
* [mbuf] End of WHOIS list.
```

5. `/msg nickname` is used to send a private message to a nickname and to start a private conversation.
6. An example list of commands from `/help` are shown below:

ADDBUTTON	ALLCHAN	ALLCHANL	ALLSERV	AWAY
BACK	BAN	CHANOPT	CHARSET	CLEAR
CLOSE	COUNTRY	CTCP	CYCLE	DCC
DEBUG	DEHOP	DELBUTTON	DEOP	DEVOICE
DISCON	DNS	ECHO	EXEC	EXECCONT
EXECKILL	EXECSTOP	EXECWRITE	FLUSHQ	GATE
GETFILE	GETINT	GETSTR	GHOST	GUI
HELP	HOP	ID	IGNORE	INVITE
JOIN	KICK	KICKBAN	KILLALL	LAGCHECK
LASTLOG	LIST	LOAD	MDEHOP	MDEOP
ME	MENU	MKICK	MODE	MOP
MSG	NAMES	NCTCP	NEWSERVER	NICK
NOTICE	NOTIFY	OP	PART	PING
QUERY	QUIT	QUOTE	RECONNECT	RECV
SAY	SEND	SERVCHAN	SERVER	SET
SETCURSOR	SETTAB	SETTEXT	SPLAY	TOPIC
TRAY	UNBAN	UNIGNORE	UNLOAD	URL
USELECT	USERLIST	VOICE	WALLCHAN	WALLCHOP

User defined commands are:

ACTION	AME	ANICK	AMSG	BANLIST
CHAT	DIALOG	DMSG	EXIT	GREP
J	KILL	LEAVE	M	ONOTICE
RAW	SERVHELP	SPING	SQUERY	SSLSERVER
SV	UMODE	UPTIME	VER	VERSION
WALLOPS	WII			

Plugin defined commands are:

UNLOAD	UNLOAD	LOAD	PY	LOAD
RELOADALL	SOURCE	TCL	RELOADALL	UNLOADALL
PL_RELOAD	RELOAD	UNLOAD	LOAD	TIMER

Type the `/HELP <command>` for more information, or `/HELP -l`

7. `/quit` is used to disconnect and exit from IRC.
8. `/join #channel` is used to join a channel. For example:

```
/join #guix
```

9. `/nick newnickname` changes your nick to *newnickname*.
Suppose you wish to move away from the computer, you can change your nick to *nick|away* or *nick|phone*.
10. `/part` is used to leave a channel.

Using rcirc

If you are using a recent GNU/Linux distribution, you should already have *rcirc* as part of GNU Emacs. You can simply start it by typing `M-x rcirc` from inside Emacs. The 'M' key represents the Meta key, which is usually mapped to the 'Alt' key. After *rcirc* connects to the IRC server, you can use `'/nick'` to change your nickname, register (only the first time) your *nick*, identify yourself, join channels, and start chatting! Since everything is a buffer in GNU Emacs, each channel is a separate buffer. For example, `#emacs@irc.freenode.net` is the `#emacs` IRC channel. All your basic buffer navigation commands will work just like they would on a file!

Some basic rcirc commands

The *rcirc* commands for the above mentioned IRC commands are given in Table 1.

Table 1

IRC	rcirc
<code>/list</code>	<code>rcirc-cmd-list</code>
<code>/msg NickServ</code>	<code>rcirc-authenticate</code>
<code>/me</code>	<code>rcirc-cmd-me</code>
<code>/whois</code>	<code>rcirc-cmd-whois</code>
<code>/msg nick</code>	<code>rcirc-cmd-msg</code>
<code>/help</code>	<code>/help</code>
<code>/quit</code>	<code>rcirc-cmd-quit</code>
<code>/join</code>	<code>rcirc-cmd-join</code>
<code>/nick</code>	<code>rcirc-cmd-nick</code>
<code>/part</code>	<code>rcirc-cmd-part</code>

~/.emacs

GNU Emacs is an extensible editor. There are a number of locations where Emacs checks for custom configurations before launching the editor. These are: `~/.emacs`, `~/.emacs.el` and `~/.emacs.d/init.el`. The start-up files can be customised, and their locations can also be changed. There are a number of ways to organise and manage your Emacs configuration. Until we get to learn Emacs Lisp, and customise Emacs as a project, we will use `~/.emacs` for all our user-specific customisation.

rcirc customisation

Create a `~/emacs.d/etc` folder in your `$HOME` directory and an Emacs file called `init-rcirc.el` in it. It should contain the following (change nick, user name and full name to suit your needs):

```
;; Default user.
(setq rcirc-default-nick "shaks")
(setq rcirc-default-user-name "Shakthimaan")
(setq rcirc-default-full-name "Shakthi Kannan")

;; Channels to join at startup.
(setq rcirc-server-alist
  '(("irc.freenode.net" :channels ("##linux-india"
    "dgpplug" "#rcirc" "#emacs"))))
```

The above is an example of Emacs Lisp code. Comments begin with two semi-colons. The `setq` construct sets the second argument value to the first argument, which is a quoted symbol. For example, the symbol `'rcirc-default-nick` is set to `shaks`. The `rcirc-server-alist` defines the initial list of channels to log in at start-up.

You can now start GNU Emacs from the GNOME terminal using the following command:

```
$ emacs -q -l ~/.emacs.d/etc/init-rcirc.el
```

You will then automatically connect to the four IRC channels.

Connecting to IRC

People join IRC channels to have their doubts regarding free and open source software clarified. Sometimes, off-topic discussions also take place. It is like live technical support, but has a social context to it. Whenever you are connected online, you must be logged in to IRC. You can have discussions in the channel, or in private, if the other party agrees. It is a good place to learn a lot about free and open source software, and you are bound to make a lot of friends. Since people from all over the world participate, which means they are online in different time zones, some channels log the discussions for future reference. As always, before asking a question, it is important for you to do your homework first. Take sufficient time and put in an effort to debug and identify the problem to the best of your ability.

Some users in the channel may ask you for more information before being able to provide you with any assistance. So, be prepared to provide all the information necessary about the bug or error when you seek help. Sometimes, people might be logged in the channel, but, they might be away from the computer. So, even if you don't get a response, be patient; come back later and ask again.

You should not paste more than four continuous lines of text in the channel, as it will 'flood' the screen for everyone else. Instead, use an external paste service like gist.github.com or

fpaste.org. These services will provide a shortened URL that you can pass around in the channel. Whoever is interested in helping you will view the contents from the link. If you enter text in the channel, it means that it is addressed to everyone in the channel. If you wish to say something to a specific user, mention their nickname first, and then type in the text.

Most IRC client software provide you with panels that list the channels that you are logged in to, and show the list of users. If someone mentions your nickname in a channel, then the corresponding channel will change colour or representation to indicate that there is a message for you. A few users are channel operators (or moderators) and they have special privileges. They are similar to 'root' users in a system, and their task is to keep the signal-to-noise ratio to a minimum, and keep a vigil on the channel.

An IRC bot is a client software that connects to the IRC server as a user, but can respond to commands. It can thus be programmed to provide many services in a channel. You can customise existing bots or write your own. Examples of IRC bots are Cerberus, Gambot and ircdd. Cinch is an example of an IRC bot-building framework written in Ruby. Bots can be used during an IRC meeting session to keep track of user questions. They can evaluate programming language constructs and return meaningful errors to newbies in the channel. They can be used to send a notification to the channel if a project test build fails, or when a new bug has been filed. The possibilities are endless.

IRC meeting protocol

A free and open source software project will have a dedicated IRC channel where the project members will meet to have discussions. Meetings can be scheduled, and can happen in different time zones depending on where the users are located. There is a protocol and etiquette to be followed during such meetings. The speaker or moderator should not be interrupted during the session.

If you wish to ask a question, type '?' and wait. When the speaker has finished and feels that you can type in your text, you will be asked to do so. After you have finished typing your content, end with 'EOF'. Similarly, if you need to speak during the session, type '!', and wait. You can give your consent or dissent to statements made in the channel using +1 or -1, respectively.

You are encouraged to read the *rcirc* manual and customise *rcirc* to your needs. If you have made it this far, do connect to *irc.freenode.net* and feel free to say 'Hi' to me. I am 'mbuf' on *irc.freenode.net*. A screenshot of an *rcirc* session is shown below:

```
19:35 <mbuf> http://ecb.sourceforge.net/screenshots/index.html
19:35 *** Arpita QUIT Client Quit
19:36 <rtnpro> !
19:36 <mbuf> rtnpro, shoot!
19:37 <rtnpro> How do we get the emacs code browser?
19:37 <rtnpro> <EOF>
```

Continued to page 100...

Are Business Intelligence Tools Really Intelligent?



Users of Business Intelligence (BI) tools have several issues to contend with. The primary concern relates to whether these tools are really intelligent and future ready. This article highlights some issues and presents a wish list for BI tools of the future.

As businesses expand and the users increase, as the volume of data grows, as databases change and more new software and other technologies get incorporated, it is observed that a Business Intelligence (BI) solution is unable to adapt to these changes. Of course, most BI software is more or less generic in nature with features such as report, dashboard, ad hoc, cache, security, etc. What it lacks is future-ready architecture. This either results in business needs being compromised or dropped, an expensive switch to the use of best-of-breed solutions, or a firm choosing to develop its own solution in-house or by outsourcing it. This results in a waste of money and time—searching for a new BI tool, hiring technical resources and implementing the solution.

So, why should business users adjust their requirements when, ideally, it should be the other way around?

As businesses grow, their expectations from a BI tool keep increasing and since these tools are generally not able to live up to the growing expectations, they get abandoned. This throws up an important question, 'Are BI tools really intelligent?'

An ideal case would be to have a BI tool which is future-ready and developer-friendly, and hence is flexible and extensible. With the growing and ever-changing business requirements, IT staff would then be able to accommodate the advanced requirements by adding features, adopting new technology and hence justifying the investments made. Ideally, such a BI framework should not be bound by any tool or technology limitations and be able to adapt to any sort of requirement—current or future. This would be a developers' paradise, giving them the liberty to do or create anything, and the business users' dream, since whatever they seek can be achieved without compromises. This will also reduce a firm's dependence on the BI vendor for any additional functionality

or patches/releases based on their product roadmap, since an in-house team would be able to add functionality.

Mentioned below are some of the instances wherein current BI tools do not match with their requirements, and need to provide much better flexibility. You may identify with some of these situations.

1. Can I add new data type as source?

Most BI tools support commonly used data sources, which are limited in number. If any new database type is to be added as a data source, it may not be possible without the database vendor providing the connect. Also, in case the data storage technology is different, like it is with Hadoop, one has to rely on the BI vendor to come up with a new patch or version. Such requests generally take a lot of time to be addressed by BI vendors who have shifted their focus from product innovation to sales. Wouldn't it be great if the developers themselves were able to add data sources, APIs, etc, and enhance the tool?

2. There's a new API in the market. Can I fetch data from it?

BI tools generally come with native connectors to certain popular APIs. But with the changing times and requirements, new and more relevant APIs come up. Fetching data from APIs other than the pre-installed ones may be impossible or often very difficult. In such a scenario, one may feel the BI tools available today are not very future-ready.

3. Charting options are few, limiting the usage of advanced analytics

After connecting to a database, reports or a dashboard are created. Most of the BI tools come with out-of-the-box charting options, which are limited in scope and may not suffice. Though some BI

tools do allow external integration of charts, they often forgo other functionalities such as exporting, email scheduling, etc.

These limited charting options affect companies or people who are looking for advanced functionalities, like data scientists and statisticians looking for statistical and advanced charting. A BI tool should allow charts to be integrated inside the report, dashboard, ad hoc, etc, with the capability to define inter-panel communication, input filters, etc. Also, even if integration of charts is external, other functionalities such as emailing, exporting, trigger, etc, should work well.

4. Reports and dashboards are clichéd.... but I don't have more options

Apart from the plain vanilla reports and dashboards, a BI tool should be future-ready enough for other visualisation options like infographics, what-if analysis, mash-ups, cubes, scorecards, or any other type which currently exists or might come up in the future.

5. The BI software UI looks so very alien!

Often, companies have their own products/software with certain navigation options, icons and colours, adhering to a chosen theme. With BI also being introduced to their solutions stack, wouldn't it be wonderful if the BI tool could be customised to match the design template of their existing solution stack? This would imply having the option to change the navigation, repository access, icons, content menu, colour, text, theme, file extensions, etc, of the BI software as well. Such exhaustive white-labelling capabilities can lead to a unified view of all the enterprise applications, leading to ease of branding, usage and viewing.

Currently, what most BI tools offer in the name of white labelling is changes in the header and footer design, colour and text— basically, very limited options.

6. There are so many tools I am compelled to use in a particular piece of BI software

Many of the BI tools require separate software and hardware to be used, such as the server, the designer tool, plugins, community plugins, etc. BI companies release enhancements within these, which at times, lead to compatibility issues. Here's food for thought: What if, using the browser itself, we are able to execute everything exactly in the way the BI solution is being accessed? Imagine—no more downloading heavy software, no more compatibility issues or separate purchase of tools, etc.

7. BI vendor engagement never seems to end – and so is the case with their billing!

Licensing presents complex issues. It may be based on the core, the number of users, the server, a combination of the above, or on data size. Also, there generally are separate licences for separate tools like the designer, server, plugin, etc. Sluggish performance of the solution leads to an increase in the number of cores and servers, and hence the more number of licences. Maintenance costs, development costs and renewal

costs are top-ups. Prices are not benchmarked and, in many cases, pricing is not crystal clear and often depends on the sales person and the bargain being struck.

8. Ad hoc capabilities are limited

Ad hoc capabilities allow business users to drag, drop and create their own reports and dashboards. Many BI tools are extremely limited here, not allowing or extending features to write custom scripts, add HTML, visualisation, custom calculated columns, etc.

9. Can I extend core functionality altogether?

All BI tools fail miserably in their ability to extend core functionality. BI tools are designed with the approach that one size fits all, wherein they sell only generic features. However, every client has a unique requirement. The ability to extend functionality and add features is something that could change the way people view and use BI. Examples of extending functionality could be things like an outlook plugin of BI, offline viewing, introducing new exporting options, a rule-based system, custom alerting notifications and triggers, custom business processes, etc. This could lead to a paradigm shift in the entire scope of BI. Frankly, the sky isn't the limit!

10. I wish I could define the sequence of events

An integrated workflow inside a BI tool could help in defining business processes and thus enhance capabilities. Examples of workflow could be things like: *run ETL AND create report AND mail to one set of users when value is between 0-50%, AND send it to other set of users when value is greater than 50%.*

11. So much software, so many screens

Companies generally use lots of software; so a client has to navigate through it based on requirements. Right now, we can only integrate BI charts inside other applications. It would be a real value addition if the BI tool is flexible enough to allow integration of other software inside the tool, interacts with this software too, and directly invokes its functions as well!

12. I just can't find BI resource people

Skilled resource personnel is a pressing problem in the BI domain. Resource persons are far too few and the salary they command is far too much, leading to outsourcing of the projects. Why should there be a separate set of resource persons for BI at all? Why can't BI tools be simple enough for a HTML/Java resource person to be able to work on the tool?

Do these situations sound familiar to you? Do you agree with the solutions? If you are able to connect with this situation, you may need to relook at your BI tool or BI implementation!  **END**

By: Nikhilesh Kumar Tiwari

The author runs his own enterprise in the data warehousing and BI space called Helical IT Solutions Pvt Ltd. He can be contacted at nikhilesh@helicaltech.com

An Introduction to Open Source Programming Languages

Programming is a matter of instructing a computer to perform certain tasks. Code, which is a language that the computer understands, comes in many flavours. The wide choice of coding languages can be pretty overwhelming to a newbie. The author presents an overview of open source programming languages in this article.

Today, programming or coding is not as big a deal as it used to be a few years back. Earlier, this was a method to solve critical problems residing in laboratories or research centres. With advances leading to ease of learning, even a school child can code today, and we have seen considerable enthusiasm from all age groups in learning how to program or code.

With the emergence of open source programming languages, one is not forced or bound to learn a single language but one can choose from a whole lot of options and can pick something that suits one best. Let me first clarify what the term *open source* actually means. As per the definition given by the Open Source Initiative Certification, “Open source is a computer program or source code available to end users free of cost and they are free to change it accordingly to make it more useful and error-free. Rather than a single proprietary ownership, it will motivate more bug-free and useful code for everyone to use.”

A few rules that a programming language must conform to in order to be declared as an open source programming language are listed below:

- Source code must be open and accessible
- Derived works should also remain open source
- Free redistribution
- Integrity of the author's source code should be maintained
- Licence must not restrict other software
- No discrimination against fields of endeavour

How open source works

Open source shows the source code to end users and gives them full freedom to change or modify it as per



their needs. The final edited version is again available for the target audience but will invariably be more effective and error-free by now.

Here are some popular open source programming languages. Of course, these are general suggestions, but there could be some better solutions if your specific requirements are known.

Getting started with HTML and CSS

If you have no coding experience but wish to learn how to code, then you should opt for HTML and CSS.

Rather than a programming

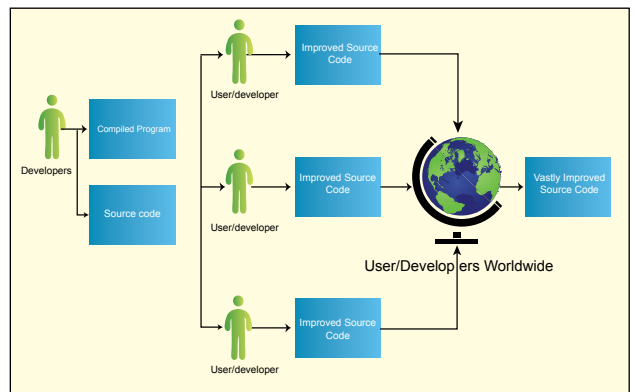
language, HTML is a mark-up language. With the help of HTML you will be able to talk to the Web browsers, and you will be embedding text, images and links into your code which a browser can understand and then give you a response. HTML code contains tags which are identifiers of the various things that you have put in your code. A simple snippet of HTML code looks like what follows:

```
<!DOCTYPE html>
My First Heading
<html>
<body>
My first paragraph.
<h1>My First Heading</h1>
<p>My first paragraph. </p>
</body>
</html>
```

On the left we have sample HTML code with the proper syntax and tags, and on the right hand side we have the response returned by the Web browser.

	Pros	Cons
1	Development and implementation cost is low	Support and maintenance cost may be hidden in the initial package
2	More easy data transferability	Being open to all, code may be more vulnerable to the hacker community
3	Potential for fast cycle time of releases and bug fixes	Open source solutions may require additional development to enable integration with an existing proprietary environment
4	No limit on usage or target audience	Introduction of new programs/ software may require staff retraining to enable them to use open source solutions
5	Suitable for rapid prototyping and experimentation	Those considering using and developing open source 'in-house' must ensure that they have the right level of expertise to manage it effectively
6	Opportunities for customisation and innovation	Open source solutions may require additional development to enable integration with an existing proprietary environment

Designers who want some back-end programming to test their applications can rely on Ruby and Python. Both are dynamic and object-oriented programming languages with very easy



OSS

All Categories

Usage Percentage

Language	Usage Percentage (%)
C	29.3%
C++	29.2%
Java	10.2%
PHP	7.4%
Python	5.7%
JavaScript	4.0%
Perl	3.4%
C#	2.6%
Objective-C	1.9%
Shell	1.9%

Languages

Usage Percentage (%)

www.OpenSourceForU.com | OPEN SOURCE FOR YOU | OCTOBER 2015 | 99

limited device compatibility and screen variations, so it is easier to build an app which is compatible enough to run on several devices.

Build desktop applications with Visual Basic

For interacting with the desktop environment, Visual Basic and VBScript are what people have been using for years, but their limitation is that they lack fancy user interfaces. Recently, Python and Ruby have evolved as powerful yet compatible programming languages for desktop application development.

Learn basic coding with Scratch

If you don't want to be bothered with serious programming terms and concepts before creating something relevant, then you should start with the Scratch tool. This was first introduced in the US and, in the classroom, kids were engaged to teach code. The use of zombies, the Angry Birds and other cartoon characters has made it more entertaining and motivating to code. The tool is a simple plug-and-play utility with predefined behaviour, and you need to assemble it to get the behaviour required.

Figure 2 gives the statistics of the popularity of various programming languages among the user community (source: *Linkedin.com*).

Although it is said that you don't need to have any project in mind before learning programming, the truth is that if you are working on an actual project, you will be more motivated

Table 2

Language	Popular compilers
PHP	Wamp, Zamp, Lamp
Java	NetBeans, Eclipse
Python	PyCharm
Ruby	ColdRuby

to learn programming in order to achieve the desired goal. If you have certain requirements beforehand, then you can easily correlate them with the learning environment and work towards what needs to be accomplished.

Talking about compilers

Every code is incomplete without being compiled to give you the desired output. The compiler changes the lines of code into useful and meaningful output. Different languages use different compilers to suit and support their environment. Shown in Table 2 are the popular compilers for open source programming languages. 

By: Meghraj Singh Beniwal

The author has a B. Tech in electronics and communication, is a freelance writer and an Android app developer. He is currently working as a systems engineer at Infosys, Mysore. He can be contacted at meghraisng01@rediffmail.com

Continued from page 95...

```
19:37 <mbuf>   rtnpro, 1. Need to install "ecb" from your
distro package manager
      2. you could have searched this on the Internet :)
19:38 <rtnpro> It is not in my distro
19:38 <sumitc> !
19:38 <mbuf>   rtnpro, and which distro are you using?
19:38 <rtnpro> Its Fedora 9
19:39 <rtnpro> I have got emacs but not emacs code browser
19:39 <mbuf>   rtnpro, you can always install from source
19:39 <techno_freak> rtnpro, http://ecb.sourceforge.net/
downloads.html
19:39 ***      khushbu QUIT Ping timeout: 244 seconds
19:39 <rtnpro> ok
19:39 <mbuf>   sumitc, shoot!
19:39 <sumitc> what is a tag-file?
19:40 <rtnpro> What factors should decide the choice of our
editor?
19:40 <mbuf>   rtnpro, wait!
19:40 ***      pushkal JOIN
19:40 <mbuf>   sumitc, the TAGS file contains the details of
the reference count,
      and locations of variables/functions et. al.
```

```
19:41 <sumitc> So, a tag file is always associated with a
specific file?
19:41 <mbuf>   sumitc, no, it can have information of files
in a directory
19:41 <sumitc> ok
19:41 <sumitc> <eof>
19:42 <mbuf>   sumitc, think of it as a database that
answers all your queries
      regarding code references
```



References

- [1] Freenode; <https://freenode.net/>
- [2] rcirci manual; https://www.gnu.org/software/emacs/manual/html_mono/rcirc.html
- [3] IRC Help; <http://www.irchelp.org/>
- [4] Fedora project - How to use IRC; https://fedoraproject.org/wiki/How_to_use_IRC

By: Shakthi Kannan

The author is a free software enthusiast and blogs at shakthimaan.com.

“BUSINESS MOBILITY IS CHANGING THE CORE BUSINESS PROCESSES”



According to the IDC Asia/Pacific (excluding Japan) Enterprise Mobility 2013 Market Sizing Forecast, the total addressable enterprise mobility solutions industry, which includes applications, devices, security, middleware and professional services, is expected to grow from US\$ 22 billion in 2015 to US\$ 26.7 billion by 2017. The market in India is also slated to grow from US\$ 1.7 billion in 2015 to US\$ 2.3 billion in 2017. This is where companies like VMware see a huge opportunity. Indian organisations are paying close attention to the future of business processes in the mobile-cloud era, taking India to the cusp of the next major technology wave in the Asia Pacific region. With a new generation of smartphone-powered workers, who have easier access to end-user devices and network connectivity, businesses are prioritising and reorienting themselves around mobile innovation, apps and services. Business mobility is clearly the way to go. **Diksha P Gupta** from **Open Source For You** spoke to **Sanjay Deshmukh**, general manager, business mobility, **VMware, APJ**, about the trend called business mobility and why it is important for businesses to adopt it. Excerpts:

Q Although the potential of business mobility sounds exciting, there are only a handful of companies in the field, as of now. The trend seems to be picking up at a rather slow pace.

The trend of business mobility has only just begun. It was not even an idea a decade back. We have just got started. Most of the customers we talk to have started adopting this idea more for individual productivity. They have started with allowing employees to access emails on devices like mobile phones and tablets. This increases individual productivity. Some companies have gone a step ahead and have started offering collaboration apps; but yes, I agree that only a handful of them have truly adopted business mobility. So there is a difference in the adoption styles.

The way we differentiate between them is that true business mobility means you change your business processes and you start thinking ‘mobile’ first. Retail is a classic demonstration of how to serve customers better and make employees more effective. That is where business mobility is being embraced. But that percentage is very small today. In our survey, we found only 15-17 per cent of customers have truly embraced business mobility, which means they have changed business processes to adopt business mobility. The reason that number is so low is that we have just got started. We do believe that a lot of customers will transition to adopting business mobility 100 per cent in a short period of time. Further, our survey

reveals that even though the percentage of companies that have adopted business mobility so far is so small, everybody wanted to get there. So it is not a question of whether to do it or not, but when to do it.

Q So, when is the time appropriate for any business organisation to adopt business mobility?

Business mobility is a journey. You can start with a small thing like email and productivity apps, and then you get on to the bigger ones. Different businesses have different triggers. We have often seen that millennials, as a group of people, are faster adopters of business mobility. Because that is the way they like to engage and work. For companies, the trigger comes from a business transformation initiative, when there is pressure from the business side to drastically change their customer experience and the way they interact with them. So we have had banks, for example, adopting this concept. In India, the most popular initiative we are seeing is tab banking, where the bank is coming to the consumer either to capture the KYC (know your customer) details or to offer other services. Every bank in India will have to take these initiatives since this will become the default way of engaging with the customer. Some of the banks have started early, but every one will gradually do it.

If you look at what is required to enable business mobility, you need a mobile device, an application and a platform like ours, to secure the device and to deliver the application. So, those are some of the initiatives that will trigger the adoption of business mobility. Speaking of tab banking again, it truly qualifies as a business mobility initiative because it has changed the business process. Earlier, the customer used to come to the bank to provide the details and get things done. Now the banks bring the services to customers, which has changed the experience completely. So this is a classic example of how business mobility is changing the core business processes, and improving service and the customer experience.

Q VMware propagates business mobility as the 'brave new model of IT'. Can you elaborate on this concept?

The need for the 'brave new model of IT' arises because the business landscape is changing. It is in a state of flux and is changing dynamically. You cannot continue to do what you have been doing over the last 25 years. That is where the need for the new model of IT comes in, because the way we worked in the past 25 years, whether it was with the client server or the mainframe,

will not take you into the cloud era to give you the benefits that these new technologies can. The new model of IT, as defined by us in one line, is 'One cloud, any application, any device'.

So it all starts with having one cloud. This means that whether you have a data centre or you are consuming services from a managed or public cloud, it's important that there is a common architecture and that it's all built on software defined architecture. Customers, in such cases, can shift their workload wherever they want to. So, if they want to increase their capacity, they can always leverage the public cloud. And conversely, if they want to reduce capacity, they can go back to a private cloud. This is possible only when there is a common architecture across all these environments. That's how the concept of one cloud starts. It can run on any hardware—regular or converged. All this is done to deliver applications, which the consumers can use from any device.

Q Is the concept of business mobility restricted only to enterprises or can SMEs also leverage its advantages?

This is for everyone. If you look at our customer base, we have OYO Rooms on the list. OYO Rooms is not among the top 1000 companies in India, but it leverages business mobility. We have a lot of such customers

"The need for the 'brave new model of IT' arises because the business landscape is changing. It is in a state of flux and is changing dynamically. You cannot continue to do what you have been doing over the last 25 years."

who have realised that they need to leapfrog ahead. They are competing with the bigger enterprises. Even a small company is competing for a consumer who has the option of going to a large enterprise. So both large and small enterprises are competing in the same space. They want to leapfrog ahead with the help of technology, so that they can offer a better, differentiated service to the customers. In short, business mobility is a big opportunity for SMEs who can truly transform their businesses with it.

Q What are the sectors that you are looking at for business mobility?

Business mobility is relevant for everyone. Having said that, some sectors will adopt it faster than the others. Retail is the sector adopting business mobility the fastest, followed by finance, education, healthcare, manufacturing, government, and so on. These are the leading sectors.

Q Business mobility looks like the way to go but enterprises still seem to be stuck in the past. What are the reasons for this?

First, they are worried about how to manage all of this, where every employee will be coming in with their own

devices. The second concern is data security and how to keep all the information secure. The third concern is the overall security of the devices, in terms of the information that they carry. So basically, security and manageability are the two major concerns and roadblocks stopping customers from embracing this technology. Wherever we are able to engage in these conversations with the customers, and convince them on how we will address their security concerns, they come on board. We try to tell that with business mobility, the users do get flexibility yet the enterprise remains as secure as is possible.

Q What are the top three considerations that a company must look at while considering business mobility as a solution?

The top three considerations from the technology stand point include heterogeneity, simplicity and having everything running on a secure infrastructure. What we tell our customers is that when you are making a transition to business mobility, make sure that these three aspects are the core design elements in your strategy. Heterogeneity is important, because if you build your strategy for business mobility adoption based on one application, one device or one platform, it will defeat the purpose. So you need to be prepared that whatever solution or platform that you are building supports any device—not only the devices available in the market today but also the devices that will come out in the future.

The second element is consumer grade simplicity. Enterprises generally have functionalities but what

they lack is simplicity. SAP as an application can do everything. It's not that SAP lacks functionality. What people struggle with when they use it, is its lack of simplicity. So user-friendliness is of prime importance.

Last, but not the least, all the applications need to run on a trusted infrastructure, because if you don't keep an infrastructure perspective while building all this, which is your cloud and data centre infrastructure, you will not be able to leverage all the investments and innovations that you have done in that space. These are the key elements to be kept in mind to enable business mobility.

Q What are the roadblocks that one can encounter while adopting business mobility?

If you build applications that align with these three key elements, the possibilities are that you may not have to face any roadblocks. But if you miss out on any one of these, the move to business mobility may not be as fruitful for the company as it should be. For instance, let's say the heterogeneity element is missed out, and someone takes the application-centric approach. After the application is operational, one may feel that the current infrastructure is good for that application but may not be suitable to run other applications. So, this needs to be checked at an early stage. Similarly, one cannot restrict an application to just one platform, like iOS or Android. There are so many platforms and so many devices available today, which needs to be considered before devising a business mobility strategy. **END** 

Read more stories on Components in
www.electronicseb2b.com

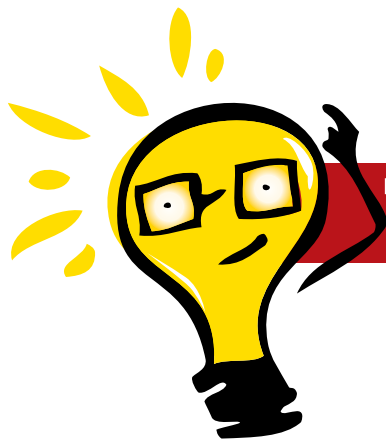
TOP COMPONENTS STORIES

- The latest in power converters
- India's leading component distributors
- Growth of Indian electronics components industry
- The latest launches of components for LEDs
- The latest launches of components for electronics

**ELECTRONICS
B2B
INDUSTRY IS AT A**

CLICK OF A BUTTON

Log on to www.electronicseb2b.com and be in touch with the Electronics B2B Fraternity 24x7



TIPS & TRICKS



How to find the largest files/directories

Here is a tip on how, by using a combination of commands, you can find the five largest files:

```
find . -type f -print0 | xargs -0 du | sort -n | tail -5 |  
cut -f2 | xargs -I{} du -sh {}
```

To find the five largest directories, use the following command:

```
find . -type d -print0 | xargs -0 du | sort -n | tail -5 |  
cut -f2 | xargs -I{} du -sh {}
```

—Gaurav Kumar, gk3eee@gmail.com



Simulating the `wc -l` command using `sed`.

Many tasks can be done with the `sed` command. The following code will simulate the `wc -l` command which counts the number of lines present in the file:

```
[mickey]$ sed -n '$ =' file.txt
```

16

Before demystifying the above command, let's understand how the `sed` command works. `sed` reads a line from the input file into a pattern buffer (the internal buffer used by `sed`), applies commands (if any) on the pattern buffer and finally prints the modified line on the standard output stream.

Here, the `'='` command prints the line number followed by its contents. `'$ ='` prints the last line number and its contents and the `'-n'` option suppresses the default printing of the pattern buffer. Hence, it displays only the last line number – now, isn't this interesting?

—Narendra Kangralkar,
narendrakangralkar@gmail.com



Checking a laptop's battery status via the command line

Have you ever tried to check your laptop's battery status from the command line? If not, here's how to go about it.

GNU/Linux provides the `'acpi'` command which can be used to check this.

Execute the following commands to install the `'acpi'` utility in Ubuntu:

```
[bash]$ sudo apt-get update  
[bash]$ sudo apt-get install acpi
```

Now, execute the `'acpi'` command. Shown below is the output when the adapter is not connected to a power source:

```
[bash]$ acpi -b  
Battery 0: Discharging, 96%, 02:41:08 remaining
```

You get the following output when the adapter is connected to a power source:

```
[bash]$ acpi -b  
Battery 0: Charging, 95%, 00:07:37 until charged
```

—Narendra Kangralkar,
narendrakangralkar@gmail.com



HTTrack: Browse a website without the Internet

This is a GUI based software that downloads the entire website from the Internet for future reading.

For example, if there's a text based tutorial website available to users when the Internet is connected, you can use this software to copy the website once to browse further on your local computer.

To install this software, run the following command:

```
#sudo apt-get install httrack
```

The software will be installed. Now go to the terminal and type the following command:

```
#httrack
```

...and then fill in the project name, website URL

and destination folder where a copy of the website will be stored.

—**Mohammed Rampurawala**,
mohammed.rampurawala@hotmail.com



No need to rewrite the command

Sometimes we forget to write *sudo* at the beginning of a command and then have to rewrite the whole command again, starting with *sudo*. To avoid this, you need to use “!!” after *sudo* and it will put the last command after *sudo*.

As an example, the following command...

```
#apt-get install gcc
#sudo !!
```

...will execute as follows:

```
#sudo apt-get install gcc
```

—**Dilip Pandit**, panditdilip@gmail.com



View manual pages while using Vim Editor

To be able to read the man pages while working on a file in Vim Editor, you do not need to come out of Vim. Just bring the cursor on the word and press K (capital), and you will be taken to the man page associated with the word.

To go to a specific section (e.g., system calls, second section) of the manual page, bring the cursor onto the word and press 2K; it opens the second section of the manual pages for that word.

—**Raveendra Reddy Padala**,
raveendrapadala@gmail.com



Rename and re-size images

Fond of your new camera but can't put up with the terrible names of the images? Do you also want to prepare these for publishing on the Web? A simple bash script is what you need:

```
#!/bin/sh
counter=1
root=mypict
resolution=400x300
for i in `ls -1 $1/*.jpg`; do
    echo "Now working on $i"
    convert -resize $resolution $i
    ${root}_${counter}.jpg
    counter=`expr $counter + 1`
done
```

Save the script in a file called *picturename.sh* and make it executable with the following command:

```
chmod u+x picturename.sh
```

Then store it somewhere in your path. Now, if you have a bunch of *.jpg* files in the directory */path/to/pictdir*, all you have to do is execute the following command:

```
picturename.sh /path/to/pictdir
```

...and in the current directory, you'll find *mypict_1.jpg*, *mypict_2.jpg*, etc, which are the re-sized versions of your original pictures. You can change the script according to your needs, or, if you're just looking for super-simple image resizing, try looking at the *mogrify* command with its geometry parameters.

—**Albert Levay**, chrishlove55@gmail.com



Know the temperature of your hard disk in Linux

You can read your hard disk's temperature if the disk has an inbuilt temperature sensor. *hddtemp* is a tool that will help you to read the temperature. It will give you the temperature of your hard drive by reading the Self-Monitoring Analysis and Reporting Technology (S.M.A.R.T.) information on drives that support this feature. Only modern hard drives have a temperature sensor.

Step 1: Install *hddtemp* by typing the following command:

```
#yum install hddtemp
```

...or whatever your package manager supports.

Step 2: In the terminal, type the following command:

```
#hddtemp /dev/sda
```

Here *'/dev/sda'* is the hard disk that we are trying to get the temperature of.

—**Bhargab Choudhury**,
bhargabchoudhury24@gmail.com

Share Your Linux Recipes!

The joy of using Linux is in finding ways to get around problems—take them head on, defeat them! We invite you to share your tips and tricks with us for publication in *OSFY* so that they can reach a wider audience. Your tips could be related to administration, programming, troubleshooting or general tweaking. Submit them at www.opensourceforu.com. The sender of each published tip will get a T-shirt.

DVD OF THE MONTH

Here are some useful Linux distributions for advanced users.



Kali Linux 2.0

Here is a Debian based Linux distribution aimed at advanced penetration testing and security auditing. Kali contains several hundred tools for various information security tasks including forensics and reverse engineering. This is the most significant release of Kali since 2013 and comes with a new 4.0 kernel. The bundled DVD has a 64-bit live edition.

antiX 15

This is a fast, lightweight and easy-to-install Linux CD distribution based on Debian Testing for Intel-AMD x86 compatible systems. antiX offers users an environment suitable for old computers. You can find the ISO image in the folder *other_isos* on the root of the DVD. The bundled ISO image is for 32-bit systems.

VectorLinux 7.1 'Light'

Here is a full-featured system with multimedia, office and networking capabilities, featuring IceWM as the desktop environment. VectorLinux is known for its speed, performance and stability. The 'Light' edition of VectorLinux comes with a small memory footprint and light applications. You can find the ISO image in the folder *other_isos* on the root of the DVD. The bundled ISO image is for 64-bit systems.

What is a live DVD?

A live CD/DVD or live disk contains a bootable operating system, the core program of any computer, which is designed to run all your programs and manage all your hardware and software.

Live CDs/DVDs have the ability to run a complete, modern OS on a computer even without secondary storage, such as a hard disk drive. The CD/DVD directly runs the OS and other applications from the DVD drive itself. Thus, a live disk allows you to try the OS before you install it, without erasing or installing anything on your current system. Such disks are used to demonstrate features or try out a release. They are also used for testing hardware functionality, before actual installation. To run a live DVD, you need to boot your computer using the disk in the ROM drive. To know how to set a boot device in BIOS, please refer to the hardware documentation for your computer/laptop.



January 11th-13th, 2016 • Bengaluru



Electronics
For You expo
Introducing... Changing... Manufacturing... Solutions



January 11th-13th, 2016 • Bengaluru

For more information, talk to us at +91-11-40595505 or email at grommybiz@etv.in



EDB
ENTERPRISEDB

DO MORE WITH POSTGRES

**IT HERO:
YOU**
FIND BUDGET
FOR NEW PROJECTS



EnterpriseDB - a Magic Quadrant Leader for Operational Database Management Systems*

Advanced Technology

Reduce risk and boost performance

Handle More Workloads

Tackle more projects

Huge Savings

Cut budget by 80% or more

EnterpriseDB Software India Private Limited Unit #803, 8th Floor Godrej Castlemaine Next to Ruby Hall,
Sassoon Road Pune - 411 001 India | Tel + 91-20-30589500/01 | Fax + 91-20-30589502 | EnterpriseDB.com

* The Gartner report, Magic Quadrant for Operational Database Management Systems, by Donald Feinberg, Merv Adrian and Nick Heudecker, was published October 16, 2014.